$2.50 IN U.S.A.
WINTER 1979 VOLUME 1 NUMBER 3

A McGRAW-HILL PUBLICATION

onComputing™

GUIDE TO PERSONAL COMPUTING

Computer Games for 1980

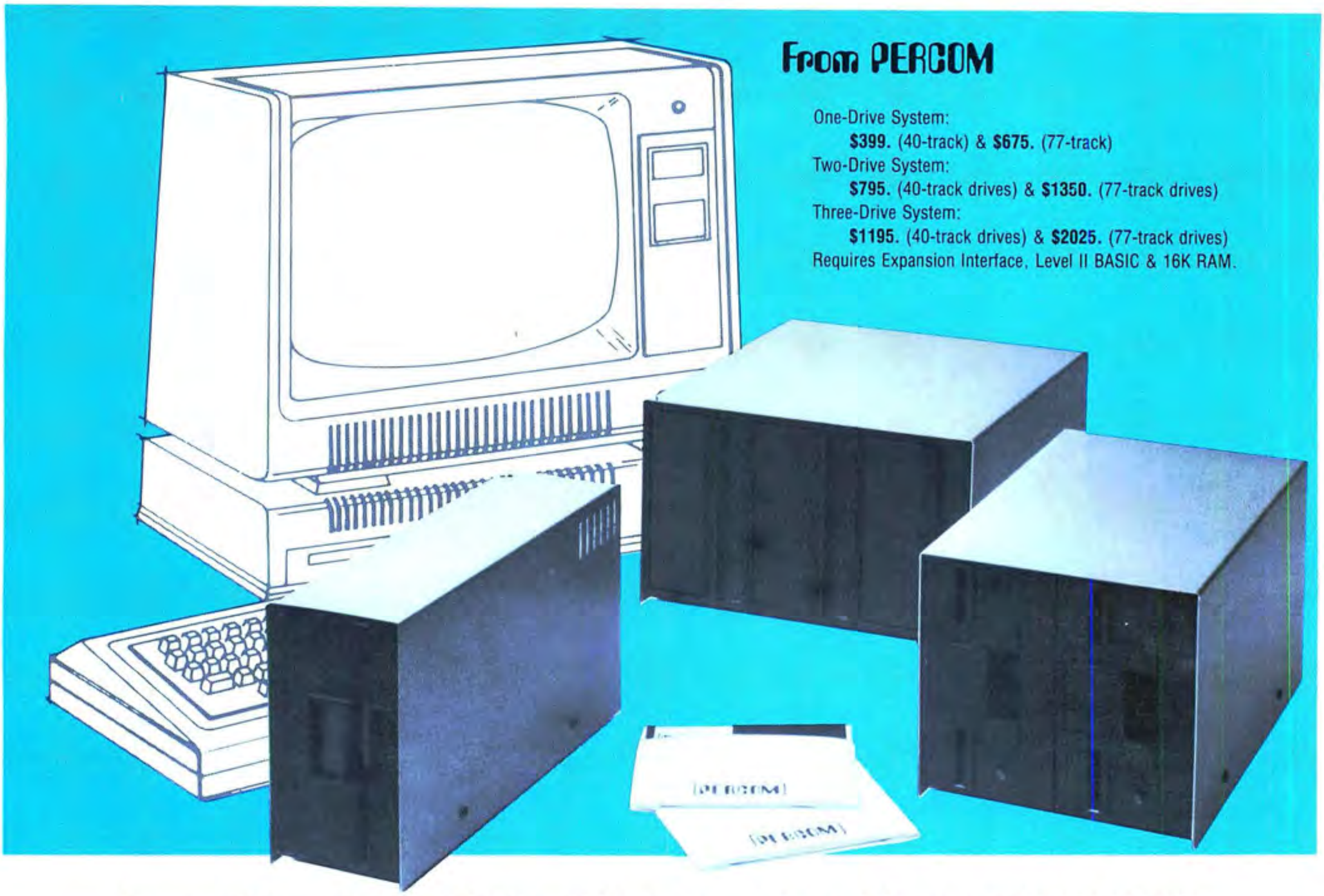
Sargon vs. Microchess: Which Plays Better Chess?

Teaching Science with Personal Computers

Understanding Software: What You Need to Know

The Heathkit H8 Computer

Winter 1979 onComputing Volume 1 Number 3



Low Cost Add-On Storage for Your TRS-80*. In the Size You Want.

When you're ready for add-on disk storage, we're ready for you. Ready with six mini-disk storage systems — 102K bytes to 591K bytes of additional *on-line* storage for your TRS-80*.

- **Choose either 40-track TFD-100™ drives or 77-track TFD-200™ drives.**
- **One-, two- and three-drive systems immediately available.**
- **Systems include Percom PATCH PAK #1™, on disk, at no extra charge. PATCH PAK #1™ de-glitches and upgrades TRSDOS* for 40- and 77-track operation.**
- **TFD-100™ drives accommodate "flippy disks." Store 205K bytes per mini-disk.**
- **Low prices. A single-drive TFD-100™ costs just $399. Price includes PATCH PAK #1™ disk.**
- **Enclosures are finished in system-compatible "Tandy-silver" enamel.**

Whether you need a single, 40-track TFD-100™ add-on or a three-drive add-on with 77-track TFD-200™s, you get more data storage for less money from Percom.

Our TFD-100™ drive, for example, lets you store 102.4K bytes of data on one side of a disk — compared to 80K bytes on a TRS-80* mini-disk drive — and 102.4K bytes on the other side, too. Something you can't do with a TRS-80* drive. That's almost 205K bytes per mini-disk.

And the TFD-200™ drives provide 197K bytes of on-line storage per drive — 197K, 394K and 591K bytes for one-, two and three-drive systems.

PATCH PAK #1™, our upgrade program for your TRSDOS*, not only extends TRSDOS* to accommodate 40- and 77-track drives, it enhances TRSDOS* in other ways as well. PATCH PAK #1™ is supplied with each drive system at no additional charge.

The reason you get more for less from Percom is simple. Peripherals are not a sideline at Percom. Selling disk systems and other peripherals is our main business — the reason you get more engineering, more reliability and more back up support for less money.

In the Product Development Queue . . . a *printer interface* for using your TRS-80* with any serial printer, and . . . the *Electric Crayon*™ to map your computer memory onto your color TV screen — for games, animated shows, business displays, graphs, etc. Coming PDQ!

™ TFD-100, TFD-200, PATCH PAK and Electric Crayon are trademarks of PERCOM DATA COMPANY.
*TRS-80 and TRSDOS are trademarks of Tandy Corporation and Radio Shack which have no relationship to PERCOM DATA COMPANY.

PERCOM

PERCOM DATA COMPANY, INC.
DEPT. B • 211 N. KIRBY • GARLAND, TX. 75042

To order add-on mini-disk storage for your TRS-80*, or request additional literature, call Percom's toll-free number: 1-800-527-1592. For detailed Technical information call (214) 272-3421.

Orders may be paid by check or money order, or charged to Visa or Master Charge credit accounts. Texas residents must add 5% sales tax.

Percom *'peripherals for personal computing'*

Circle 110 on inquiry card.

Solve your personal energy crisis. Let VisiCalc™ Power do the work.

With a calculator, pencil and paper you can spend hours planning, projecting, writing, estimating, calculating, revising, erasing and recalculating as you work toward a decision.

Or with VisiCalc and your Apple* II you can explore many more options with a fraction of the time and effort you've spent before.

VisiCalc is a new breed of problem-solving software. Unlike prepackaged software that forces you into a computerized straight jacket, VisiCalc adapts itself to any numerical problem you have. You enter numbers, alphabetic titles and formulas on your keyboard. VisiCalc organizes and displays this information on the screen. You don't have to spend your time programming.

Your energy is better spent using the results than getting them.

Say you're a business manager and want to project your annual sales. Using the calculator, pencil and paper method, you'd lay out 12 months across a sheet and fill in lines and columns of figures on products, outlets, salespeople, etc. You'd calculate by hand the subtotals and summary figures. Then you'd start revising, erasing and recalculating. With VisiCalc, you simply fill in the same figures on an electronic "sheet of paper" and let the computer do the work.

Once your first projection is complete, you're ready to use VisiCalc's unique, powerful recalculation feature. It lets you ask "What if?," examining new options and planning for contingencies. "What if" sales drop 20 percent in March? Just type in the sales figure. VisiCalc instantly updates all other figures affected by March sales.

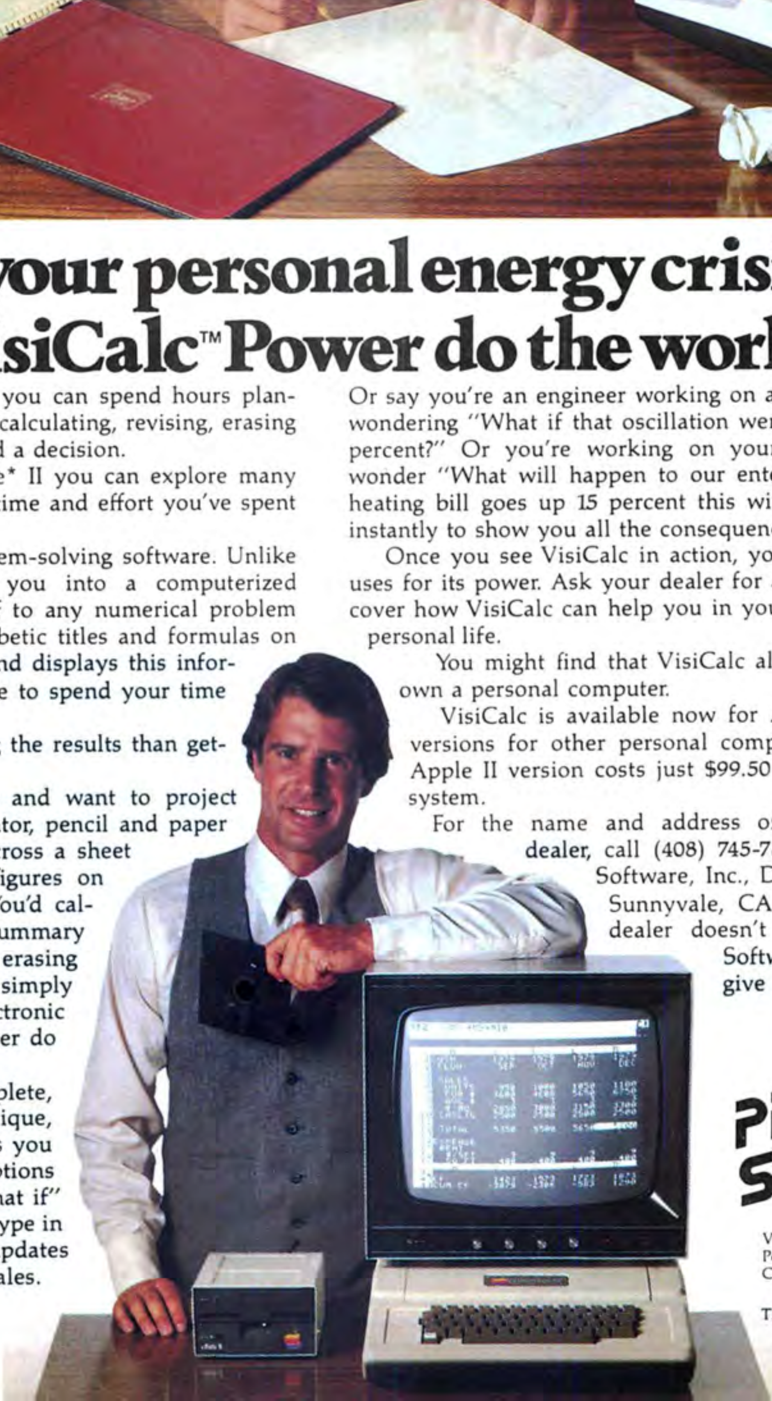
Or say you're an engineer working on a design problem and are wondering "What if that oscillation were damped by another 10 percent?" Or you're working on your family's expenses and wonder "What will happen to our entertainment budget if the heating bill goes up 15 percent this winter?" VisiCalc responds instantly to show you all the consequences of any change.

Once you see VisiCalc in action, you'll think of many more uses for its power. Ask your dealer for a demonstration and discover how VisiCalc can help you in your professional work and personal life.

You might find that VisiCalc alone is reason enough to own a personal computer.

VisiCalc is available now for Apple II computers with versions for other personal computers coming soon. The Apple II version costs just $99.50 and requires a 32k disk system.

For the name and address of your nearest VisiCalc dealer, call (408) 745-7841 or write to Personal Software, Inc., Dept. O, 592 Weddell Dr., Sunnyvale, CA 94086. If your favorite dealer doesn't already carry Personal Software products, ask him to give us a call.



PERSONAL SOFTWARE

VisiCalc was developed exclusively for Personal Software by Software Arts, Inc., Cambridge, Mass.

TM–VisiCalc is a trademark of Personal Software, Inc.

*Apple is a registered trademark of Apple Computer, Inc.

onComputing

THE PERSONAL CHECK WRITER — *Andrew Recupero* — **59**
Your computer can print your checks.

Product evaluation: HEATHKIT H8 — *Mark Snow* — **65**
A sixteen-year-old boy reviews this popular personal computer.

DEPARTMENTS

Editorial	4	Letters	80
Article Policy	30	Glossary	83
Book Reviews	32, 55, 58	New Products	86
Profile of Sol Libes	68	Coming Events	95
Computer Store Directory Update	70	Reader Service	96
		interAction	96

NATIONAL ADVERTISING SALES REPRESENTATIVES:
HAJAR ASSOCIATES INC

East:
280 Hillside Av
Needham Heights MA 02194
(617) 444-3496

521 Fifth Av
New York NY 10017
(212) 682-5844

Midwest:
664 N Michigan Av
Suite 1010
Chicago IL 60611
(312) 337-8008

West, Southwest:
1000 Elwell Ct
Suite 227
Palo Alto CA 94303
(415) 964-0706
or (714) 540-3554

onComputing is published four times a year by onComputing Inc, 70 Main St, Peterborough NH 03458, a wholly-owned subsidiary of McGraw-Hill, Inc. Address all mail except subscriptions to above address: phone (603) 924-7217. Address all subscriptions, change of address, USPS Form 3579, and fulfillment questions to onComputing, POB 307, Martinsville NJ 08836. Second class postage paid at Peterborough NH 03458 and at additional offices. (ISSN 0194-8075) Subscriptions are $8.50 for one year in USA and its possessions. In Canada and Mexico $10 for one year. Other foreign countries $12 for one year surface delivery. Single copy price is $2.50 in the USA and its possessions, $2.90 in Canada and Mexico, $4 in Europe, and $4.50 elsewhere. Foreign subscriptions and sales should be remitted in United States funds drawn on a US bank. Printed in the United States of America.

Address all editorial correspondence to the editor at 70 Main St, Peterborough NH 03458. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of onComputing.

Subscription WATS Line:
(800) 258-5485

Volume 1
Number 3
Winter 1979

FEATURES

COMPUTER GAMES AND TOYS — *Chris Morgan* — **6**
The latest crop of microcomputer games for the holiday season.

USE YOUR POCKET CALCULATOR TO SAVE MONEY — *Louis Hohenstein* — **12**
An inexpensive way to budget your money at home and business.

UNDERSTANDING PERSONAL COMPUTER SOFTWARE — *Louis E Frenzel, Jr* — **17**
A look at the "invisible" side of your computer.

SARGON VS MICROCHESS — *John Martellaro* — **26**
Two chess-playing programs battle it out on your video screen.

A SIMPLE CASSETTE DIRECTORY — *John K Appleton* — **31**

THE TALCOTT MOUNTAIN SCIENCE CENTER — *Daniel Barstow* — **34**
Kids learn about science with personal computers in Connecticut.

A VOICE FOR BILL — *Mark Dahmke* — **48**
Personal computers can help the handicapped to "speak."

Cover photo by Ed Crabtree

Editorial onComputing

Playing Games

We were in a dungeon the other day. Poison gas filled the room to our left. To our right lay the uncharted reaches of the lower dungeon where gold and precious jewels were hidden. A giant snake guarded the corridor before us. We fought off an evil necromancer and a horde of giant spiders, when suddenly a specter came toward us. We ran for the stairway, but it was too late! One of our party was carried off. Then...I could go on, but I don't want to scare you.

An unpublished Edgar Allan Poe story, perhaps? No, it's just my personal computer creating a simulated world of adventure.

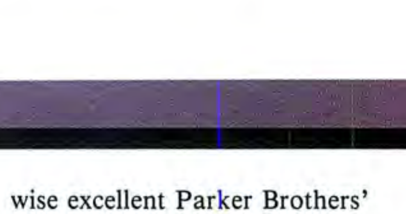
In this issue of onComputing we provide a brief roundup of new computer games and gadgets for 1980 (see page 6). These devices fall into two different categories. The first is the dedicated unit that contains a microprocessor designed to perform a specific function, such as Texas Instruments' Language Translator. Its built-in computer is used exclusively to generate words through a loudspeaker. The second way to play computer games is to use a general-purpose personal computer and plug in a game program available on cassette or floppy disk. An inexpensive alternative to commercial computer-game programs is to use the programs printed in personal-computer magazines. A thorough index to such games can be found in *Belais' Master Index to Computer Programs in BASIC*, available from Falcon Publishing, 140 Riverside Av, POB 688, Ben Lomond CA 95005.

The Marketplace

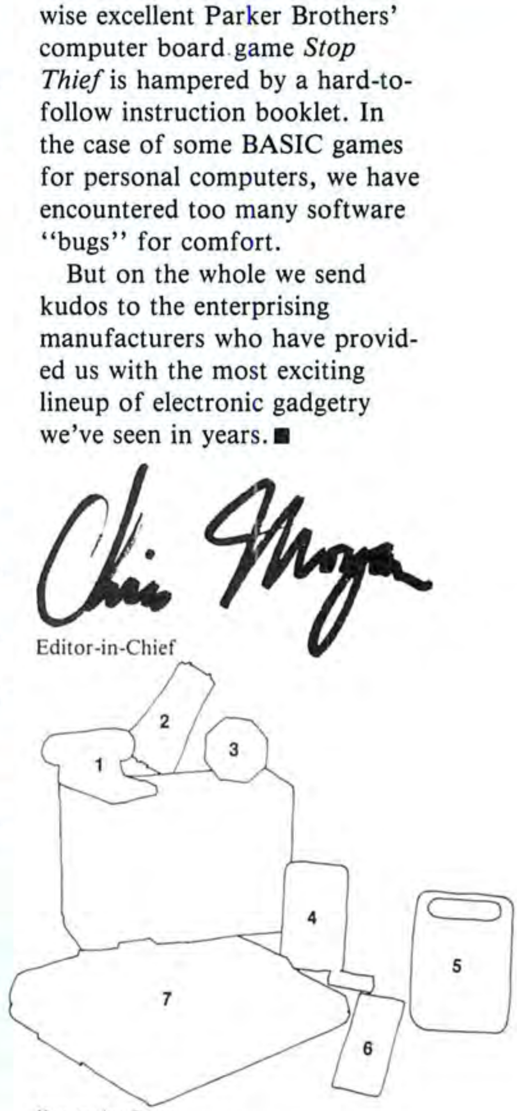
Who buys computer games? *Toy and Hobby World* (a trade magazine for the toy industry) recently reported the results of a survey conducted by NPD Research Inc examining the buying habits of Americans. Not surprisingly, parents do the bulk of the buying. They're willing to spend the high prices for electronic games, and most often buy for the older children in the family or for themselves.

What do they get for their money? Quite a bit, all things considered. The onComputing staff served as eager consultants for the evaluation of the games (*quite* eager, in fact). We are impressed with the industry's latest offerings. My only criticism is that, in the rush to get products to the marketplace, there have been some loose ends. For example, the otherwise excellent Parker Brothers' computer board game *Stop Thief* is hampered by a hard-to-follow instruction booklet. In the case of some BASIC games for personal computers, we have encountered too many software "bugs" for comfort.

But on the whole we send kudos to the enterprising manufacturers who have provided us with the most exciting lineup of electronic gadgetry we've seen in years. ■



Editor-in-Chief



Key to the Cover:
1. Scrabble Brand Sensor electric word game. 2. UFO Master Blaster. 3. Copycat electronic skill game. 4. Entex Electronic Baseball 5. Texas Instruments Speak & Spell 6. Texas Instruments Language Translator 7. Mattel Electronics Intellivision personal computer.

Circle 5 on inquiry card. →

MORE COLOR. MORE SOUND. MORE GRAPHICS CAPABILITIES.

Compare the built-in features of leading microcomputers with the Atari personal computers. And go ahead, compare apples and oranges. Their most expensive against our least expensive: the ATARI® 400™.

Start with graphics capabilities. The ATARI 400 offers 128 color variations. 16 colors in 8 luminance levels. Plus 29 keystroke graphics symbols and 8 graphics modes. All controlled from a full 57 key ASCII keyboard. With upper and lower case. And the system is FCC approved with a built-in RF modulator. That's just for openers.

Now, compare sound capabilities. Four separate sound channels and a built-in speaker. With the optional audio/digital recorder, you can add Atari's unique Talk & Teach™ Educational System cassettes.

Here's the clincher: Solid state (ROM) software. For home management, business and entertainment. Or just plug in an Atari 10K BASIC or Assembler language cartridge and the full power of the computer is in your hands.

Memory? 8K expandable to 16K. And that's just for the ATARI 400 at a suggested retail of only $549.99.

The ATARI® 800™ gives you all that and much more.

User-installable memory to 48K. A full-stroke keyboard. With a high-speed serial I/O port that allows you to add a whole family of smart peripherals. Including up to four individually accessible disk drives. And a high speed dot-matrix impact printer. And, the Atari Program Recorder is included with the 800 system. Suggested retail price for the ATARI 800 (including recorder) is $999.99.

Make your own comparison wherever personal computers are sold. Or, send for a free chart that compares the built-in features of the ATARI 400 and 800 to other leading personal computers.

ATARI®

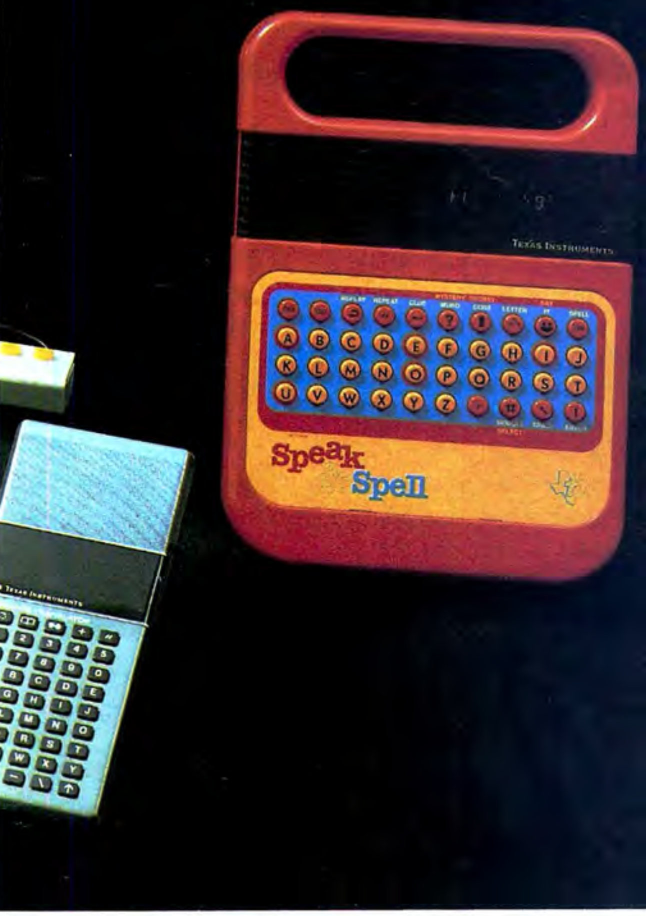
PERSONAL COMPUTER SYSTEMS

1265 Borregas Ave. Dept. C, Sunnyvale, California 94086. Call toll-free 800-538-8547 (in Calif. 800-672-1404) for the name of your nearest Atari retailer.


Co
Ga
Letters
Position
Player One
Player Two
TIGER
TRINITRON
HOME
VISITOR
10:44
:00
SONY
NFL FOOTBALL
INTELLIVISION
baseball

mputer
mes and Toys

What's New for 1980?



by Chris Morgan

Photos by Ed Crabtree

I took a stroll through my local toy store the other day. It looked like the prototype lab for an electronics company. Formerly commonplace plastic diversions were now staring at me with light-emitting diodes (LEDs), and I discovered that if I wanted to play with my new toy dump truck, I would have to learn how to program it.

There's no doubt about it — microcomputers have invaded the toy industry, and toys will never again be the same. Actually, computer games have been around for a couple of years. Early in the game, Mattel Electronics brought out a series of highly successful, handheld games including football and baseball. However, it was Simon that caught the public's fancy and paved the way for the avalanche of products that followed.

Introduced by Milton Bradley in mid-1978, Simon was an instant hit. The unit is a round plastic game of skill. It resembles a flying saucer with large colored buttons and sound effects. The idea is to try to reproduce the sequence of lights (and corresponding sounds) that are generated by the built-in microprocessor by pressing buttons. (The microprocessor, incidentally, is made for Milton Bradley by Texas Instruments.) Such was the initial popularity of the game that people formed lines to buy it, and supplies were exhausted by mid-December of last year. At about the same time, Parker Brothers introduced Merlin, a computer-controlled game resembling a telephone headset, and offering a variety of games from tic-tac-toe to blackjack.

Today the field is wide open, and computer games are being offered from literally dozens of sources. Some use microcomputers in highly sophisticated ways; others use them marginally or for trivial effects.

Milton Bradley has again come up with one of the most interesting items, Microvision. This handheld game has a square, liquid-crystal display, and can accept a total of seven different game cartridges. One game, called Blockbuster, challenges the player to knock down rows of blocks with a bouncing ball that ricochets from a paddle that is controlled in a manner similar to a large arcade game. The other game cartridges available for Blockbuster make use of the liquid-crystal array to good effect. The design is an attractive alternative to the video screen, although it takes awhile to get used to the "per-

sistence" effect of liquid-crystal displays (ie: images die out slowly, leaving short duration afterimages). This can sometimes be confusing to the eye.

Not to be outdone, Parker Brothers has introduced Stop Thief, a microcomputer-controlled, crime-detection board game. The game begins with an audible alarm from the computer. Clues are given to the players in the form of sound effects, electronically generated sounds of breaking glass, squeaking floorboards, and running footsteps. A digital display simultaneously records the building or street number where the crime is taking place. Successful sleuths are rewarded with the sounds of police sirens and gunfire. Another Parker entry is called Wildfire, an electronic handheld pinball machine that simulates the lights and sound effects of the traditional pinball machine.

Chafitz has introduced what is probably the most expensive ($2500) dedicated computer game to date: Aristotle, the microcomputer-controlled backgammon game. The luxuriously appointed unit was designed by Dr Hans Berliner of Carnegie-Mellon University and Paul Magriel, the 1978 world backgammon champion. According to Magriel, Aristotle plays an "excellent" game. I tried my hand at beating it at the Consumer Electronic Show this past summer—it certainly played a good game against me! I feigned an important appointment when I saw the inevitable coming. Aristotle beat the current world backgammon champion this past July by a score of 7-1 in a seven point match.

An updated version of the Chafitz Boris chess computer has also been introduced. It recently achieved a class B rating in the Paul Masson chess tournament—an impressive feat.

One of the most fascinating computer games to appear in the past year is Texas Instruments' Speak and Spell, a sophisticated talking game that teaches children how to spell. A microcomputer-synthesized voice asks the child to spell a word by pressing keys on a keyboard. As each key is pressed, the corresponding letter appears on an LED display. If the word is spelled incorrectly, a second request is made. A running score is kept and announced after ten words.

The unit also features several other games. Although designed for children, this unit has fascinated the adults I have shown it to. A new plug-in module called Super Stumpers is now available for the unit. This unit helps children build their spelling, pronunciation, and reading skills. It is aimed at the junior high school market.

Several familiar games have reappeared under electronic guise this year. Invicta Plastics is marketing a handheld computerized version of MasterMind, a popular game of deduction in which players attempt to guess a hidden code generated by the computer. Selchow and Righter are marketing the Scrabble Brand Electronic Word Game, the object of which is to guess your opponent's word in the fewest turns.

New Personal Computers

Several personal computers are being introduced this fall, and all of them are aimed squarely at the video games market. Manufacturers are banking on a consumer "crossover" to these new units that combine the best features of color video games with the programmability of personal computers.

Atari has introduced two new personal computers (see "Atari's New Hybrid Computers" in the Summer 1979 onComputing), the models 400 and 800. Both accept game cartridges, but can also be programmed in BASIC. Mattel's Intellivision is a 2-part system; customers can buy a video game console and later add a keyboard console to do their own programming. A wide variety of software is scheduled to be available.

Texas Instruments has introduced their new personal computer which comes with its own

Photo 1: The Scrabble Brand Sensor *is the first entry into the computer games market from a venerable institution in the field. The game is actually not like Scrabble at all, but more like Word MasterMind. Player 1 secretly enters a word into the machine. Player 2 then tries to guess the identity of the word by offering another word of the same number of letters for comparison. The machine then tells player 2 how many letters the two words have in common, and whether any of these letters are in the same position as those in the mystery word.*

The price of the game is rather high when you consider that the unit has no facility for displaying letters and words, and that the game could easily be played with pencil and paper alone.
Unit: Scrabble Brand Sensor
Manufacturer: Selchow and Righter
Price: $39.95

Photo 2: Stop Thief *is an unusual combination of board game and computer game. Players attempt to track down a thief who secretly moves around the game board. This is done by receiving visual and audible clues from a calculator-like device and moving one's detective pieces accordingly. The game bears a visual resemblance to the long-established Parker Brothers' game of Clue, but the mechanics are quite different. We thought the game was well executed and challenging, although it might be too complicated for some younger players — especially when trying to decipher the sometimes vague instruction manual.*
Unit: Stop Thief
Manufacturer: Parker Brothers
Price: $39.95

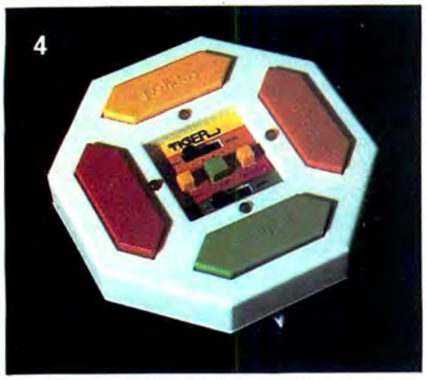
Photo 3: The UFO Master Blaster *comes as close as you can get to having a video display in a handheld computer game — with the possible exception of Milton Bradley's Microvision. The Master Blaster sports two columns of LED displays which are used to create the illusion of rocket ships traveling through space — and it works!*

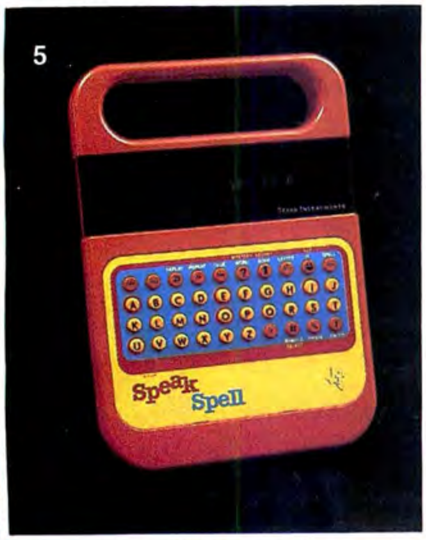
The object of the game is to shoot down as many enemy flying saucers as possible (for which you get points), while preventing them from destroying your home base station.

The player has control of both the launching and positioning of rockets, and the game is a true test of dexterity.

The game's designers also get high marks for the display, which creates a believable illusion.
Unit: UFO Master Blaster
Manufacturer: Bambino Inc
Price: $39.95

Photo 4: Copycat *is one of several imitations of Milton Bradley's highly successful game, Simon. Players attempt to duplicate progressively longer sequences of colored lights that are accompanied by sounds. The unit does in fact do this, but we found the quality to be wanting; one of the lights was burned out in the Copycat we bought, and two of the player's stations produced a sound of the same pitch, which was confusing. With no price advantage, we prefer Simon.*
Unit: Copycat
Manufacturer: Tiger Electronic Toys
Div Interstate Industries Inc
Price: $29.95





video monitor. Milton Bradley is producing a series of "Gamevision" cartridges for use with the TI machine. The games include Yahtzee, Hangman, and Connect Four all adaptations of existing Milton Bradley games.

Another entry from Mattel Electronics is Brainbaffler, a calculator-like module that plays word games such as Go Hang, Build-a-Word, Flash Words, and Third Degree Anagrams, along with accompanying sound effects. Mattel has also added Soccer to its list of handheld games.

Waddington's House of Games is marketing The Game Machine, programmed to play Shooting Gallery, Black Jack, Code Hunter, and Grand Prix. The unit also features a 4-function calculator and sound effects.

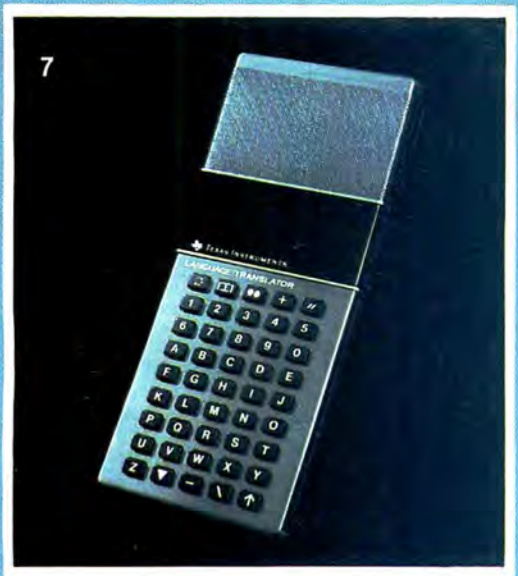
Software

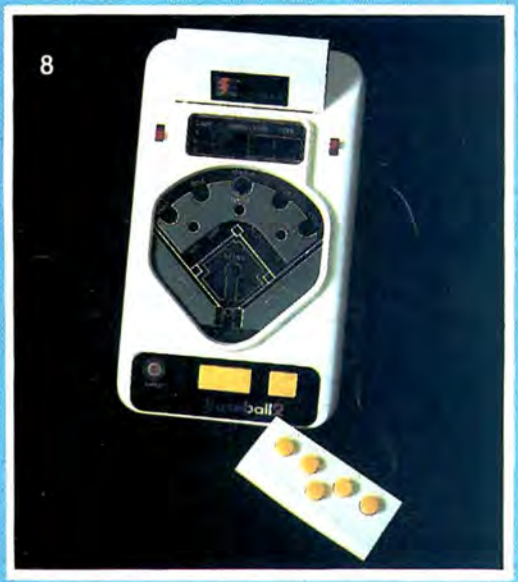
The possibilities for fun and games for personal computer owners are almost endless. Among the myriad of available game programs are such items as Time Trek for 8 K byte Pets and 4 K Level I and II Radio Shack TRS-80 computers, and Microchess for 8 K Pets, 16 K Apples, and 4 K TRS-80s, available from Personal Software, Weddell Dr, Sunnyvale CA 94086.

Several other companies manufacture excellent computer-game software, notably the GRT Corp, 1286 N Lawrence Station Rd, Sunnyvale CA 94086, and Softape, 10432 Burbank Blvd, North Hollywood CA 91601. Softape's amazing "Appletalker" program lets you play tic-tac-toe with an Apple computer, using only voice input. A cassette recorder and microphone are the only hardware required.

A new game I have particularly enjoyed is Dungeon Campaign, designed for 48 K byte Apple computers, in which players attempt to amass a fortune in gold "quadroons" while avoiding serpents, huge spiders, necromancers, and a host of other pitfalls, all housed in a complex, 4-level dungeon maze. The game is available from Synergistic Software, 5221-120th Av SE, Bellevue WA 98006. Full marks go to Synergistic for producing a very sophisticated piece of software.

Mad Hatter Software offers such items as Othello, science fiction games, and various





strategy games for the TRS-80, Apple, and other computers. The address is 900b Salem Rd, Dracut MA 01826. A visit to your local computer store (see "A Computer Store Directory" in the Summer 1979 issue of onComputing and the update in this issue for a complete list of stores) should reveal a variety of possibilities in the software area.

As far as computer games go, the sky's the limit. Ed Krakauer of Mattel Electronics predicts that electronic games sales could exceed the billion dollar mark in the not too distant future. Being a techno-freak from way back, this is music to my ears.

Well, enough typing for now. I'm going out in the backyard to program my pogo stick. Or maybe I'll add computer-controlled servo-motors to my roller skates... ■

Photo 5: Texas Instruments' Speak & Spell *is a remarkable piece of technology: the unit teaches children to spell by means of its computer-generated voice. As the child enters the letters of a word via the keyboard, they appear on an LED display. Incorrect spellings cause a verbal prompt to "try again." Speak & Spell comes with a 250-word vocabulary. Additional modules can be plugged into the unit to update its memory. During the testing of the unit, we found that adults were fascinated by it almost as much as children. Speak & Spell is one of the best examples of what the microcomputer industry can do.*
Unit: Speak & Spell
Manufacturer: Texas Instruments
Price: $50 (approximate)

Photo 6: Mattel's Intellivision: *This unexpected entry into the personal computer market comes from a company previously best known for its toys. Intellivision is a hybrid design incorporating a video game section and a separate computer module with keyboard that can be mated to the video game section; both units attach to your home color TV set.*

The unit we received for testing was an early prototype — the game module was operational, but the keyboard section was a mockup. Thus we confined our evaluation to the video game module. We found it to be well above average. In fact, the basketball game, with sound effects, is easily the most sophisticated game of its kind we've come across. If the computer module turns out to be as good as the video game section, Intellivision will be a formidable competitor in the personal computer market.
Unit: Mattel Intellivision hybrid personal computer
Manufacturer: Mattel Electronics
Price: $500 (approximate)

Photo 7: Texas Instruments' Language Translator *is a close cousin to Speak & Spell. The unit contains large-scale integrated circuitry that enables it to speak in a foreign language and translate from one language to another. Phrases, as well as sentences, can be processed through the translator. We had a chance to examine a working prototype of the translator at the Summer Consumer Electronics Show. The results were impressive indeed. The machine does a credible job of speaking French, and its audio capability puts it in front of competing units that feature LED displays only (although such units are admittedly less expensive).*
Unit: TI Language Translator
Manufacturer: Texas Instruments
Price: $250

Photo 8: The Entex Electronic Baseball *game is the best of the current crop of handheld baseball games. Enthusiasts of the sport will appreciate the sophistication of this game, and its one very unusual feature: it splits in two so that two players can compete.*

We particularly liked the pitching controls. Adroit users can pitch such exotic combinations as a curveball with knuckleball action. The batter can direct the motion of the baseball (really a string of sequential LEDs) by putting "English" on the bat. The game can also be played solo, in which case the computer does the pitching.
Unit: Entex Electronic Baseball
Manufacturer: Entex Industries
Price: $39.95

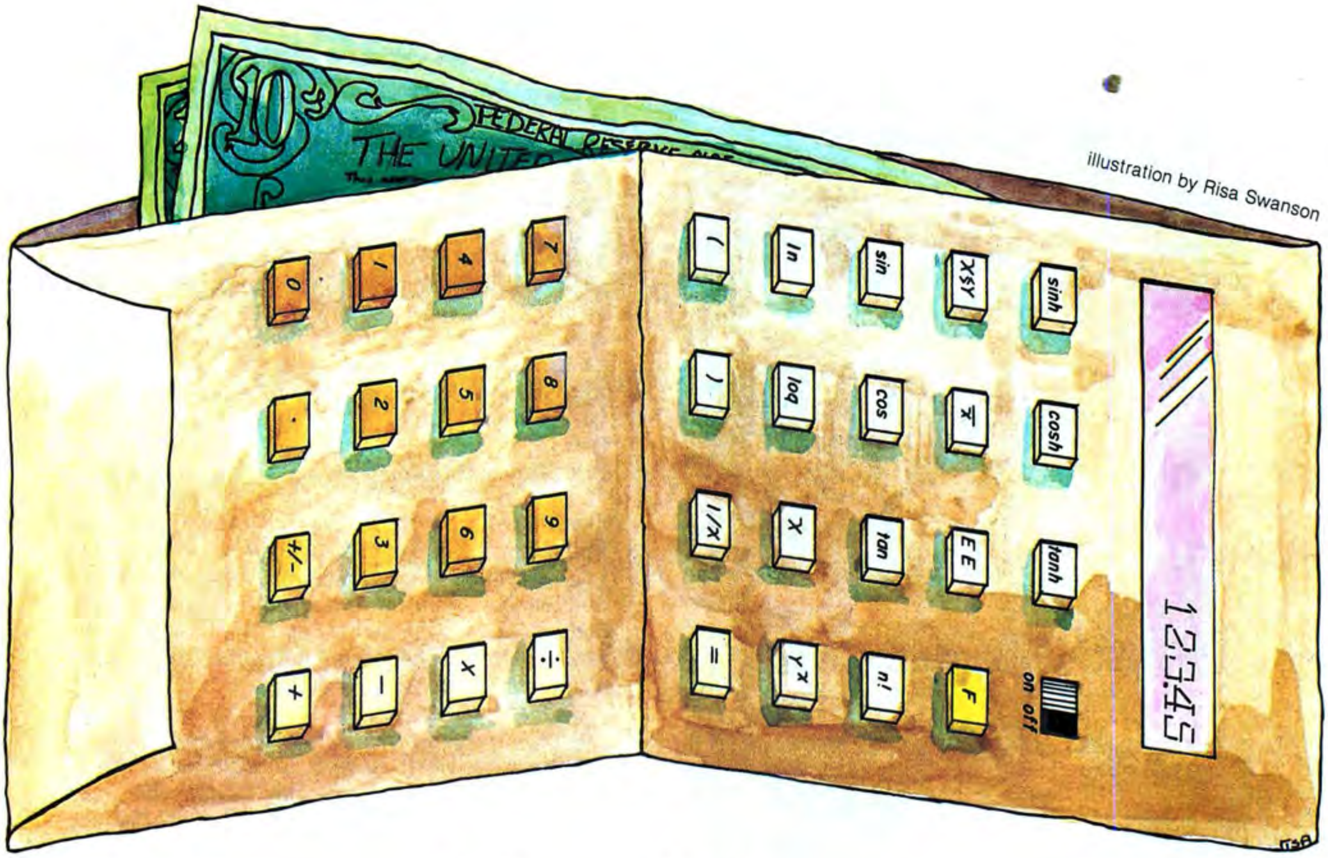


illustration by Risa Swanson

Use Your Pocket Calculator to Save Money

by Louis Hohenstein

Now there's a new way to sort bills you've paid into business expense categories. This method reduces bookkeeping time and costs and gives you more information about how your costs are running.

This new and easy method uses a programmable pocket calculator. Some of the ideas discussed herein are aimed more at the small business user, but they can be easily adapted to your home budget.

Why Classify Expenses?

If you didn't have to classify expenses, you could just add up all of your checks for expenses or purchases of goods for resale as one lump sum. However, if you run a small business or manage your household budget, you'll want to know if the money you spend each month is reasonable for each category of expense.

For example, is automobile expense reasonable? Has car maintenance risen enough so that it's time to trade? How about utilities? Are they too much, too little, or just about right? Once you know the amount you're spending, you can decide whether to hold the line, cut back, or spend more; without these figures, you can't.

So expenses are sorted into categories for expense control and effective decision-making. You must also categorize expenses for your annual tax returns.

When expenses are distributed by pencil and paper in a small business, a multicolumned accounting spread sheet is often used. When you use a spread sheet, the number of expense categories is limited to the number of columns available on the spread sheet. This in turn limits the expense information you get and your ability to control costs.

Post Office Boxes

A modern pocket-calculator memory is like the boxes used for sorting mail into pigeonholes at the post office. In the pocket calculator, each memory unit is like a different postal box — but for dollars and cents instead of mail. You can put $34.61 in box 25, and later add $57.75 more to this same box. Then the sum held in box 25 is $92.36. In the pocket calculator the *boxes*, as I've called them up until now, are called *registers*. You may add dollar amounts to any register *in any sequence* you want. This ability to add amounts to any register in any sequence permits you to add amounts from your expense checks to memory registers representing expense categories quickly and easily. (If you own a personal computer, the analogy of the post office boxes is much the same.)

As an example, to add an amount to a register:

(1)Punch the dollar amount on the calculator keyboard (such as 57.75).

(2)To add the amount to register 25, punch the keys [sum] [2] [5]. (The rectangle around the word *SUM* indicates the calculator key labeled SUM.)

If you assign 2-digit numbers to your expense categories, these numbers will correspond to the calculator's memory registers. If your expense category 25 is "automobile expense" in your *chart of accounts,* when you're finished punching in your expenses on the calculator, the total in memory register 25 is your total expense for automobile operation.

Developing a Chart of Accounts

Figure 1 shows a typical chart of accounts. You can see there are many unused expense category numbers. Use this as a guide to develop your own chart of accounts for yourself. Then as you write each check, also write the expense category number for each expense item. This is an easy way to prepare for your expense distribution later. For the small business, voucher checks with their extra accounting copy (figure 2) are a perfect way to record the expense category numbers for easy use later. Recording expense category numbers as you write each check eliminates the need to sort things out later.

Getting Expense Totals

When you're finished punching each expense amount and the corresponding expense category number for a month (or any other accounting period), you're ready to have the calculator summarize the totals. There are two ways to do this: with and without an optional printer. Without the optional printer, you can look into each calculator register corresponding to your expense categories for the total. Do this by punching the recall button (labeled RCL) and then a register number (example: [RCL] [2] [5] for automobile expense). The amount in that register will flash on the display of the calculator. You can then write the amount and go on to the next total you want.

A modern pocket-calculator memory is like the boxes used for sorting mail into pigeonholes at the post office.

The Optional Tape Printer

When a printer is used with your calculator, the one-at-a-time inquiry for the totals in each register is eliminated along with all writing. I suggest you use the printer. It's well worth the cost of less then $200.

When you're finished posting expenses, tell the calculator to print each sum in the registers. To do this, you simply push three buttons: [INV] [2nd] [LIST]— and the printer prints out the totals in each register along with the register number.

Figure 3 is an example of a listing of the expense totals and the corresponding register numbers. For sixty expense categories (a large number for most applications), the automatic printing of the expense totals takes about 10 seconds.

The printer can also be set to print *each expense* amount as you're entering it, along with the expense category number. This provides a record if a keying error is made. To do this requires a short control program that I'll explain in a moment. When the calculator/printer is used to print each entry, at the end of the expense distribution process you have (1) expense totals for each category printed on the tape, (2) a record of each detail entry, and (3) a grand total of all expenses entered.

How to Use This Method

Step 1 — Set up a chart of expense accounts. You need a standard list of expense categories regardless of how you do your bookkeeping, and possibly you already have one. You can use the list of frequently used accounts shown in figure 1 along with account numbers as a start. Add or delete accounts for your special needs. Each account has a 2-digit number for use with the calculator. (Actually, any 2-digit number between 00 and 99 may be used. This provides a maximum of 100 account classifications. If you don't need this many, number your accounts from 00 to 59. This saves a few steps later, as I'll show.)

After you have set up expense category numbers, write each expense number on your checks (or in your checkbook) as you pay expenses. Whoever does your expense distribution later will then have the proper account code when he or she distributes your expenses. Remembering the expense category numbers soon becomes automatic and you'll not need to refer to the expense category numbers on your chart of accounts.

If you use a voucher check like the one shown in figure 3, write your expense numbers on the voucher section. The extra accounting copy of this check provides an ideal document as the source of

Figure 1: Typical Business Expense Categories

Your Account No.	Expense Category	Your Account No.	Expense Category
11	Salary Manager	53	
12	Salaries Partners	54	
13	Salaries Officers	55	
14	Salaries Administrative	56	
15	Salaries Office	57	Insurance
16	Salaries Selling	58	
17		59	
18	Bonuses	60	
19		61	
20	Commissions	62	Interest
21		63	
22		64	
23	Advertising	65	Legal and accounting fees
24		66	
25	Auto and truck	67	
26		68	Miscellaneous
27		69	
28		70	Office supplies
29		71	
30		72	
31		73	
32	Collection expense	74	Postage
33		75	Printing
34	Contributions	76	Rent
35		77	
36	Delivery expense	78	Repairs and maintenance
37		79	
38		80	Salesman expense
39		81	Small tools and equipment
40		82	Supplies
41		83	
42		84	Taxes and licenses
43		85	Taxes, payroll
44	Dues and subscriptions	86	
45		87	
46		88	
47		89	
48	Entertainment	90	Telephone
50	Equipment rental	91	Travel
51		92	Utilities
52			

accounting data, and speeds month-end expense distribution.

Before assigning the numbers to your accounts, read the section on calculator alternatives and decide first which you'll use, since this affects the numbers you assign.

The Programmable Calculator

There are several models of pocket calculators with adequate memory. Several are made by Hewlett-Packard (models 19, 67, and 97). Others are the Texas Instruments' models 58 and 59. The calculator I'll use as an example is the Texas Instruments 59 (TI-59).

This calculator was priced at slightly under $250 at this writing and you can use it as it comes off the shelf from the store, or with a control program.

Optional step 2a — Buy a calculator and use it as it comes off the shelf. As soon as you turn on the TI-59 (with or without the printer), it blinks on with sixty memory counters (00-59) ready to go. You can start right away and add each expense amount to a counter by entering the dollar amount and then pushing $\boxed{\text{SUM}}$ $\boxed{2}$ $\boxed{5}$ (or any other register number as appropriate). You can increase the number of counters to 100 (00-99) by pushing several buttons for more counters, as described in the TI-59 instruction book. This procedure is *memory repartitioning*; it's simple enough to

Text continued on page 74

Software Authors—see separate ad on page 120.

onComputing™

Name (Please print)

Street/Apartment Number

City | State/Province/Country | Postal Code

Please start my subscription with: ☐ Winter 1979 ☐ Spring 1980

☐ U.S. 1 yr. (4 issues) @ $8.50 ☐ Canada & Mexico 1 yr. (4 issues) @ $10.00

FOREIGN (to expedite service, please remit in U.S. funds drawn on a U.S. bank.)
☐ Europe (and all other countries, except above) 1 yr. @ $12.00—surface delivery

☐ Bill VISA ☐ Bill Master Charge ☐ Bill me (North America only)

Card No.

Expiration

Four digits above name – Master Charge only

Signature Date

Please allow eight weeks for processing. Thank you **4139**

onComputing™

Name (Please print)

Street/Apartment Number

City | State/Province/Country | Postal Code

Please start my subscription with: ☐ Winter 1979 ☐ Spring 1980

☐ U.S. 1 yr. (4 issues) @ $8.50 ☐ Canada & Mexico 1 yr. (4 issues) @ $10.00

FOREIGN (to expedite service, please remit in U.S. funds drawn on a U.S. bank.)
☐ Europe (and all other countries, except above) 1 yr. @ $12.00—surface delivery

☐ Bill VISA ☐ Bill Master Charge ☐ Bill me (North America only)

Card No.

Expiration

Four digits above name – Master Charge only

Signature Date

Please allow eight weeks for processing. Thank you **4139**

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS Permit Number 45 Martinsville NJ 08836

POSTAGE WILL BE PAID BY ADDRESSEE

onComputing
POB 307
Martinsville NJ 08836
USA

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS Permit Number 45 Martinsville NJ 08836

POSTAGE WILL BE PAID BY ADDRESSEE

onComputing
POB 307
Martinsville NJ 08836
USA

Understanding Personal Computer Software

by Louis E Frenzel Jr

All personal computers are made up of two main parts: hardware and software. Hardware is the visible part of the system: the computer and its peripherals. Software is the invisible part. Even though it is unseen, it is the software that lets the hardware do useful work. Hardware and software are equally important, but in a sense, a knowledge of hardware operation is irrelevant to the use of a computer. If you buy a commercial computer, you do not need to know how the central processing unit (CPU), terminal, or floppy disk works in order to use the computer. However, you do have to know how to use the software—otherwise nothing happens. For this reason, your real success with a computer will come more quickly if you understand the software.

The purpose of this article is to help you get the most value from your computer by introducing you to the subject of software. It provides a complete overview of software and establishes a foundation on which to build your future knowledge.

Some Definitions

The computer and its peripherals are typically referred to as hardware. Hardware includes the electronic circuits and electromechanical devices that make up the computer system. Hardware is the term used to describe any of the physical parts of a computer system. The integrated circuits that make up a computer, the video terminal, and the dot matrix printer are all types of computer hardware.

Hardware is really only half of a computer system, though. It is possible to have perfectly working computer hardware that is totally worthless. Why? Because the computer itself and its peripherals perform no useful work until the computer is given a *program*. A program is a sequence of computer instructions that tell the computer what to do. The computer instructions are listed sequentially in a special way that commands the computer to perform some useful function. Once the computer is given a program, it can execute the instructions and produce something of value. Programs are a type of software.

Software: What Is It?

All of the programs that a computer uses are collec-

Louis E Frenzel Jr manages the Educational Division at the Heath Company. This article is adapted from his forthcoming book on microcomputers to be published by Howard W Sams and Co.

tively known as software. Software is a general term used to describe any single program or group of programs. Software also includes such things as operating procedures and *documentation*. Any printed reference material such as a user's manual is also considered part of the software. Together, the software and the hardware form a complete computer system. One is no good without the other.

Software comes in a variety of forms. It is usually supplied on some form of mass storage media such as paper tape, magnetic tape, or floppy disk. To be useful, the software must be placed in the computer's main memory (programmable memory) and executed. Software can also be stored in *read-only memory* (ie: a type of non-erasable memory used for storing important information).

Software is the real interface between you and the computer. Although peripheral devices such as terminals allow you to communicate with the computer, it is actually the software that makes this communication possible. Peripheral devices are only the physical implementation of this communication. There must be some type of program in the computer that facilitates your communications with the machine. Software provides this interface between the user and the hardware. (See figure 1.)

An example will illustrate this concept more clearly. When speaking to the computer, you are not talking to

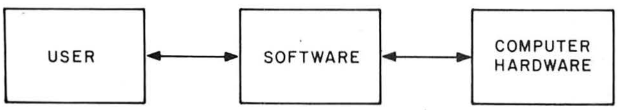


Figure 1: Software is the real interface between the user and the computer.

the hardware. Instead, you are talking to a program. Assume that the computer sends a question to the cathode-ray tube in your video terminal which asks:

WHICH PROGRAM DO YOU WISH TO USE?
A. CHESS
B. CHECKBOOK
C. BASIC
D. MAILING LIST
E. LEARNING PROGRAMS

The computer lists the programs it is capable of running. Select the desired program by pressing one of the keys A thru E on the terminal keyboard. The desired program is then retrieved, perhaps from a floppy disk, moved into main memory, and executed. This dialog takes place because of a program.

Types of Software

The software used by a computer can be classified in two specific categories: *applications* software and *systems* software. All general purpose computer systems use both types.

Applications software refers to those programs that you will use to perform some specific function. Applications software may be an inventory control program for a small business, a program that plays a game of chess on a color TV set, or a computer-aided instruction program to teach algebra. While you may develop some special applications programs yourself, you will probably buy applications programs developed by manufacturers and software companies.

Systems software consists of all of the programs, languages, and documentation supplied by the manufacturer with the computer. These are the pro-

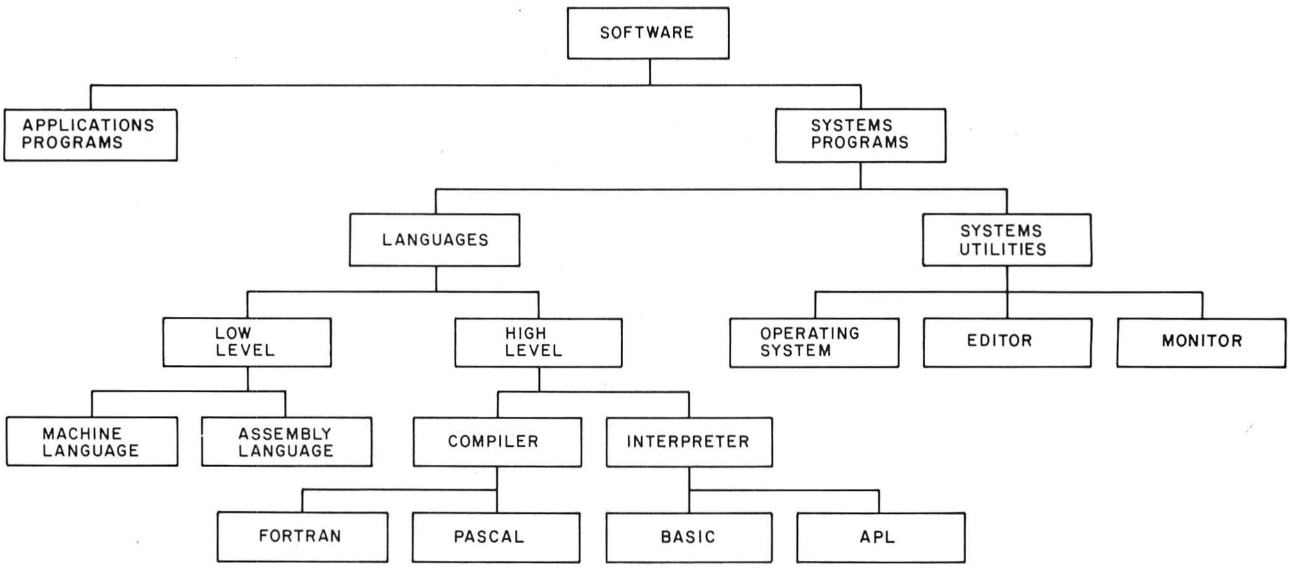


Figure 2: Interrelationships between different types of software. Applications programs are the programs you buy or write to use on your computer. Systems programs enable the computer to run applications programs in a variety of ways by means of systems utilities programs. Programs are written in both low- and high-level languages.

grams that allow you to communicate with the computer and to write or develop your own programs. Systems software facilitates the use of the computer and makes it faster, more efficient, and more convenient. This is the software that is applied in developing applications software. Despite the availability of systems software, many users do not develop their own applications software because of their lack of motivation or programming expertise.

Figure 2 is a chart showing the hierarchy of software available for microcomputers. It is divided into the two basic categories: applications software and systems software. Since applications programs are strictly left up to the programmer, they will not be further discussed here. Instead, I will concentrate on the systems software. The systems software is further divided into the categories of *computer languages* and *systems programs* or *systems utilities.*

Computer Languages

A language is a system of communications. It usually consists of all of the verbal or written symbols and expressions that are used to exchange ideas and information. A language contains all of the symbols, char-

Example Program The following is a simple, binary machine-language program that adds 5 and 9 together and stores the result in the computer's memory (see text at right).

Memory Address	Content (Instruction, Data or Address)	Meaning
00000000	00111110 (instruction)	Retrieve the first number from memory.
00000001	00000101 (data)	This is the first number to be added (5).
00000010	11000110 (instruction)	Add the second number to the first number.
00000011	00001001 (data)	This is the second number to be added (9).
00000100	00110010 (instruction)	Store the sum.
00000101	00000111 (address)	Least significant half of the 16-bit address where sum is to be stored.
00000110	00000000 (address)	Most significant half of address.
00000111	00001110 (data)	Sum

Glossary:

Address: analogous to a street address, this term refers to the number or label of one location in the computer's memory.
Instruction: One of the predefined tasks that a microcomputer can perform. Instructions are "hardwired" into the computer (ie: the user need only give the name of the instruction, and the computer will carry out the details). Machine language consists entirely of instructions from the computer's instruction set.
Hexadecimal: A type of number system based on the number sixteen, compared with the decimal system, which uses ten as its base. Each digit in a hexadecimal number represents some power of sixteen; sixteen symbols are used to make up hexadecimal numbers (1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F). The hexadecimal number system allows large numbers to be expressed in compact form, and simplifies the task of the computer programmer.
Least Significant and Most Significant Halves of a Number: The rightmost and leftmost groups of digits in a number, respectively. For example, the most significant half of the binary number 1100110011111111 is 11001100.
Octal: A type of number system based on the number eight (see hexadecimal). Each digit in an octal number represents some power of eight. A total of eight digits (0,1,2,3,4,5,6, and 7) are used to represent octal numbers.

acters, procedures, and syntax that would be used in communicating with the computer, and vice versa.

As you can see in figure 2, there are two basic types of languages used in microcomputers: *low-level* languages and *high-level* languages. These are futher divided into four categories: *machine* and *assembly* languages, and *compiler* and *interpreter* languages. Machine and assembly languages are low level, while interpreters and compilers are high-level languages. High-level languages are similar to our own native, English language. Low-level languages are closer and more compatible with the hardware of the computer. All languages are used to develop programs.

Photo courtesy Heath Company

Photo 1: The Heath H89. A typical integrated or all-in-one personal microcomputer containing video terminal with keyboard and CRT display, CPU, and floppy disk mass storage.

Machine Language

The lowest form of computer language is machine language. Machine language uses binary numbers to tell the computer what to do. A simple, binary machine-language program to perform the addition of the two numbers 5 and 9 and store the sum is shown in the box at left.

As you might expect, programming in binary is extremely time consuming and error prone. Machine language programming can be simplified by using *octal* or *hexadecimal* notation instead of binary notation (see Glossary at left). The binary machine-language program in the example reduces to a somewhat simplified form, as shown below in hexadecimal form:

Memory Address	Content (instruction, data or address)
00	3E
01	05
02	C6
03	09
04	32
05	07
06	00
07	0E

Whether you are using binary, octal, or hexadecimal notation, you are working directly with the central processing unit when programming in machine language. You have full control over all *registers* (ie: locations in a microprocessor or other computer where numerical values can be temporarily stored), *input/output ports* (ie: physical connectors on the computer that enable it to send and receive information to and from the outside world), and other features. However, the disadvantage is that you must implement each minute detail of the program yourself. The above program represents a significant amount of work to accomplish a simple task such as adding two numbers. Yet in many applications, this is desirable.

Most microcomputers can be programmed in machine language. Microcomputers used for teaching computer operation and programming are designed primarily to be programmed in machine language. They typically feature an octal or hexadecimal keyboard for entry of programs and data, and 7-segment light emitting diode (LED) displays for reading out instructions and data (see photo 2). Unless you are just learning microcomputer operation and programming, or developing very short and simple programs, you will not want to use machine-language programming. Instead, you should use a *higher-level* language that makes programming much faster and easier.

Assembly Language

Assembly-language programming was developed to overcome the disadvantages and objections of machine-language programming. Yet assembly-language programming has many of the same advantages as machine-language programming in that there is access to and control over the registers, I/O (input/output) ports, and other features of the computer. Assembly-language programming allows you to use shorthand expressions for computer instructions, addresses, and data. Instructions are referred to by *mnemonics*, which are simply two- or three-letter abbreviations for the function performed by the instruction. For example, an instruction that tells you to load the accumulator register may have the mnemonic LDA. An add instruction would have the mnemonic ADD. The instruction that tells us to store the accumulator register might have the mnemonic STA, and so on. The mnemonic is simply a shorthand way of remembering an instruction. These mnemonics are used in place of binary, octal, or hexadecimal codes when writing a program.

In addition to using mnemonics to represent instructions, decimal numbers or names can be used to refer to addresses and data. For example, you may use the name IOSUB to refer to the starting address of an I/O

subroutine. (A subroutine is a separate "program-within-a-program" that can be called on by the main program to perform some task.) The term CASH may refer to the amount of money you have on hand in your checking account. These names are used instead of the actual numbers. This allows you to refer to these values without knowing their exact values or location in memory.

Using these features, you can see that the machine-language program given earlier reduces to the much simpler and easier to understand assembly language shown below:

```
MVI    NUMB 1
ADI    NUMB 2
STA    SUM
```

The first instruction, MVI, retrieves the first number to be added NUMB 1 (5) from memory. The next instruction, ADI, adds the second number NUMB 2 (9) to the first number, producing the sum. The last instruction STA stores the sum in a memory location called SUM. The instructions MVI, ADI, and STA are the mnemonics. The names NUMB 1, NUMB 2 and SUM are memory addresses designating where the various data numbers are stored or will be stored.

Once you write an assembly-language program, you have the problem of getting the computer to understand that program. The assembly language was developed to simplify the development of the program. You can readily learn the use of mnemonics for instructions and names for addresses and data. The computer, on the other hand, cannot. It understands only its native tongue: machine language or binary numbers. How can the two be made compatible? By the use of a special translation program called an *assembler.* The assembler is a program that is loaded into the computer's memory. Its primary function is to translate your assembly-language program into machine language. Just as a foreign language interpreter will translate between two different foreign tongues, the assembler will convert your assembly-language program into its binary machine code equivalent.

In computer jargon, your assembly-language applications program is called the *source* program. The source program is the input to the assembler. The assembler then performs the translation and generates the equivalent machine code. This equivalent machine program is called the *object code.* The object code is the actual binary machine instructions and data that the computer can use (see figure 3).

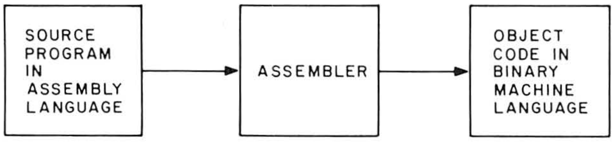


Figure 3: The assembler converts the source program into machine language.

Photo courtesy Heath Company

Photo 2: Low cost microcomputer trainer used for teaching machine language programming and interfacing uses hexadecimal keyboard and display.

Once the input source program has been translated into object machine code, the assembler is no longer needed. The newly generated object code can then be executed. In a typical computer system, the assembler is stored on paper tape, magnetic tape, or a floppy disk. When needed, the assembler can be called up from mass storage and loaded into the memory. You enter the source program via your computer keyboard. The source program is also stored in the memory. The assembler translates the source program into object code and stores it in another portion of memory. The object code or program can then be stored in some external medium. In this way, the program can be retrieved and executed when needed.

There is one important characteristic of assembly-language programming that you should be aware of. Assemblers typically translate a source program to an object program on a one-to-one basis. That is, one assembly-language mnemonic translates into one machine-code instruction. As you will see later, one statement in a higher-level language may translate to many machine code instructions.

Higher-Level Languages

Higher-level languages were designed to make programming even easier. Some knowledge of the microprocessor and its structure is required to program proficiently in assembly language. But with higher-level languages, no knowledge of computer operation or structure (the latter is also called architecture) is required. In fact, you do not even need to know binary numbers. Higher-level languages make it possible for just about anyone to program.

You may have already heard of some of the more popular higher-level languages, such as BASIC, FORTRAN, and Pascal. All of these languages can be classified as one of two different types: *compilers* or *in-*

SPECIAL DELIVERY

A totally new printer has arrived. It delivers some special advantages to small business and home computer systems. New, expanded paper-handling capabilities. Excellent print quality and long-run reliability. All these high quality features at an affordable price is indeed *special delivery.*

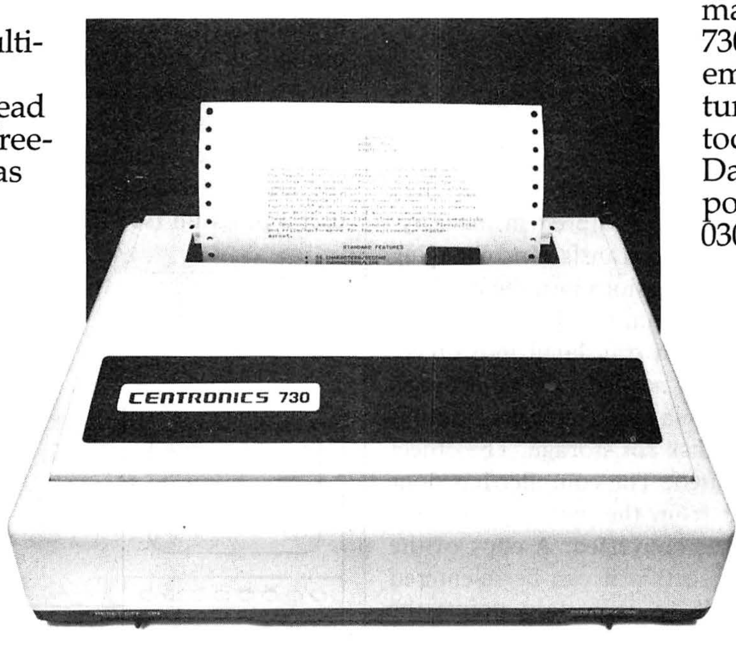
3-in-1 Paper Handling Model 730's exclusive three-in-one paper handling system gives you built-in flexibility. Payroll checks on pre-printed continuous forms. Inventory listings on fanfold. Direct mail on cut sheets. General information on roll paper. The 730 does it all.

Exceptional Print Quality Model 730's 7 x 7 dot matrix prints clear, legible characters—even when printing multiple copies. The adjustable print head utilizes the same free-flight technology as Centronics' 700 Series—the most successful line of dot matrix printers in the world—to give you longer head life and smoother operation.

Outstanding Reliability The simplicity of the Model 730's design employs fewer parts for higher reliability and less maintenance. You get more in-service time for your money.

Compact Design Model 730 packs all this performance capability into a stylish unit that can fit right on your desktop. It's no larger than a portable typewriter and weighs less than 10 pounds.

For all its features, Model 730 is priced within the budget of even the most cost-conscious systems owner. For more information on the Model 730, other products or employment opportunities, write or call today: Centronics Data Computer Corporation, Hudson, NH 03051, (603) 883-0111.



CENTRONICS PRINTERS DELIVER THE WORD

Circle 30 on inquiry card.

terpreters. Like assemblers, compilers are also translation programs. They take an input source program written in the target language and convert it into the machine code that can be executed by the computer.

Whenever you write a program in compiler language, you do not write individual lines of instructions that correspond one-to-one with instructions in the microprocessor's machine language. Instead, you will write expressions, statements, or commands that tell the computer what to do in a way that is unique to that particular language. These commands can be English-like statements, or *algebraic* expressions. For example, the addition of two numbers, A and B, can be accomplished by the simple algebraic expression $X = A + B$, which would be translated by the compiler into the machine code equivalent of two, three, or more machine instructions.

Another typical higher-level language statement is: PRINT "Annual Inventory." This English statement would be translated by the compiler into many machine-code instructions that would perform that operation. You can see how much simpler and easier programming becomes when you tell the computer what to do by simply using English or mathematical expressions.

Once a higher-level language source program is written, it must be converted into machine code in order for the computer to execute it. This, of course, is done by the compiler.

The Compiler

The compiler, like the assembler, is a program that resides on a floppy disk or other mass-storage medium. When it is needed, it is called up by the computer and loaded into memory. Your source program is then entered on the keyboard. It too is stored in memory. The compiler then performs the translation, changing the various statements and expressions into the machine code known as the object program (see figure 4).

Once your program has been translated into object code, the compiler is no longer needed. The object code which was stored in memory may also have been output to magnetic tape or floppy disk for storage. The object program can then be executed. The compiler has done its job and can be erased from the memory to make room for the program it has converted. A copy of the compiler is kept elsewhere so that it can be re-entered when needed again. This is an important point: the object program resulting from translation by a compiler (or an assembler) can be executed without those translation programs being in memory. Compilers are not needed to run the program, only to translate it.

The primary advantage of higher-level languages is that they greatly speed up and simplify the programming process. The program to solve a given problem can be written more quickly and easily in a higher-level language than in machine or assembly language. As expected, the development of higher-level languages has greatly expanded the use of computers. Familiarity with

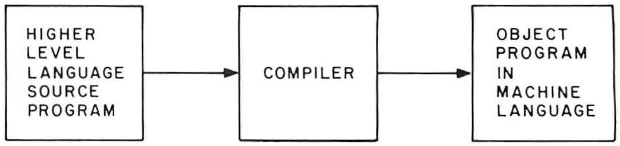


Figure 4: The compiler translates the input source program into machine language.

the hardware is no longer necessary in order to write a program. You have only to learn a simple, higher-level language that is much like your own native language. Higher-level languages allow individuals with no computer knowledge to take advantage of the many benefits the computer provides.

The Interpreter

Another type of higher-level language is the interpreter. An interpreter is a program that, unlike the compiler, stays in the memory and interprets or executes the higher-level language source program. Interpretative higher-level languages use statements, expressions, and commands that are similar to those used in compiler languages. To prepare a program, write these expressions line by line, using the unique syntax of that language to solve the problem. As usual, the program is entered into the computer by means of the terminal.

The interpreter can be stored on some type of external medium such as magnetic tape or floppy disk, and then brought into memory when it is needed. In some computers, the interpreter is stored in read-only memory. In this way it is always available when needed. The main difference between the interpreter and compiler is that the interpreter must reside in memory along with the user's program (see figure 5).

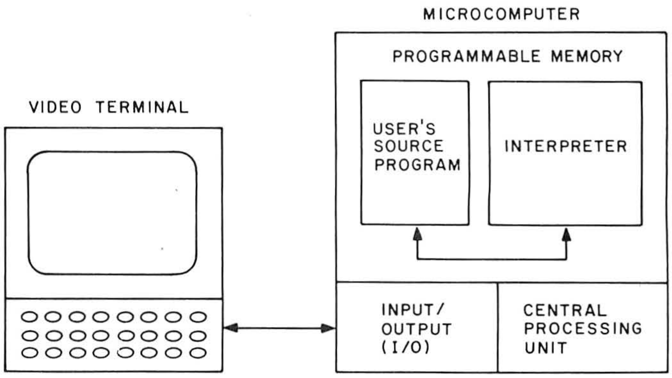


Figure 5: The interpreter remains in the computer's memory in order to translate the user's program.

The interpreter does not translate the user's source program into object code, as do compilers and assemblers. Instead, the interpreter looks at each statement or expression, then performs the desired operations by selecting appropriate subroutines within the interpreter itself that carry out the desired function.

While machine code eventually does execute the program, the interpreter does not generate it. The machine code subroutines to perform specific operations already exist inside of the interpreter. They are simply called upon as needed.

Interpreters are extremely handy and easy to use. Like compiler languages, they are easy to learn, and programs can be quickly written using English and algebra-like statements and commands. The big disadvantage of an interpreter is its slow speed. Interpreters execute programs at a speed that is ten to twenty times slower than equivalent machine code generated by an assembler or compiler. In some applications, this slow execution speed is not only undesirable, but also unacceptable. Keep in mind, however, that computers execute at a high rate of speed. Even with the inefficiency imposed by the interpreter, they are more than fast enough for most applications.

Despite this disadvantage, interpreters are widely used. They are convenient and easy to use because they are totally *interactive*. You can literally sit in front of the computer terminal, converse with the computer, and solve problems directly through the interpreter. By comparison, when a compiler language is used, the source program must first be entered and then compiled. The object code generated by the compiler is then loaded and executed. If you wish to change something in the program, such as add features or correct errors, the entire process must be repeated. The program changes are made and entered. The new source program is then recompiled. Finally, the new program is executed. With an interpreter, changes and additions can be made interactively. As soon as you discover that you need to modify or add something, you can do it immediately through the interpreter. When using an interpreter, you get the feeling that you are truly speaking to a computer and commanding its operation.

Now let us take a look at some of the more popular higher-level languages.

FORTRAN

One of the oldest higher-level languages is FORTRAN. It was developed in the late 1950s at IBM. FORTRAN means *formula translation*, which gives a clue as to its application. FORTRAN was primarily developed to speed up and simplify complex scientific and engineering calculations. It is a general purpose, higher-level language, and it can be used in a wide variety of applications. However, FORTRAN was primarily invented to optimize mathematical computation. Most higher-level languages are referred to as problem-oriented languages in that they were specifically developed to optimize the solution of a particular class of problems.

The primary language of FORTRAN is algebra. The rules and syntax of FORTRAN allow even the most complex mathematical formulas and expressions to be reduced to simple, single-line statements. English-like

Text continued on page 77

By John Martellaro

Sargo Micro

Which Plays Better

Here's a sobering thought: your personal computer may be able to beat you at chess! Such an idea was unthinkable just a few years ago, but today's chess playing programs can give you a run for your money — at least if you're a beginner.

Two chess programs presently available for the Apple II and TRS-80 personal computers are Sargon and Microchess. Sargon, written by Dan and Kathe Spracklen (and implemented on the Apple II by Gary Shannon), is available from Hayden Book Co Inc, 50 Essex St, Rochelle Park NJ 07662, \$19.95 on cassette. Microchess 2.0 for the Apple II (Microchess 1.5 for the TRS-80), written by Peter Jennings, is available most notably from Personal Software, 592 Weddell Dr, Sunnyvale CA 94086.

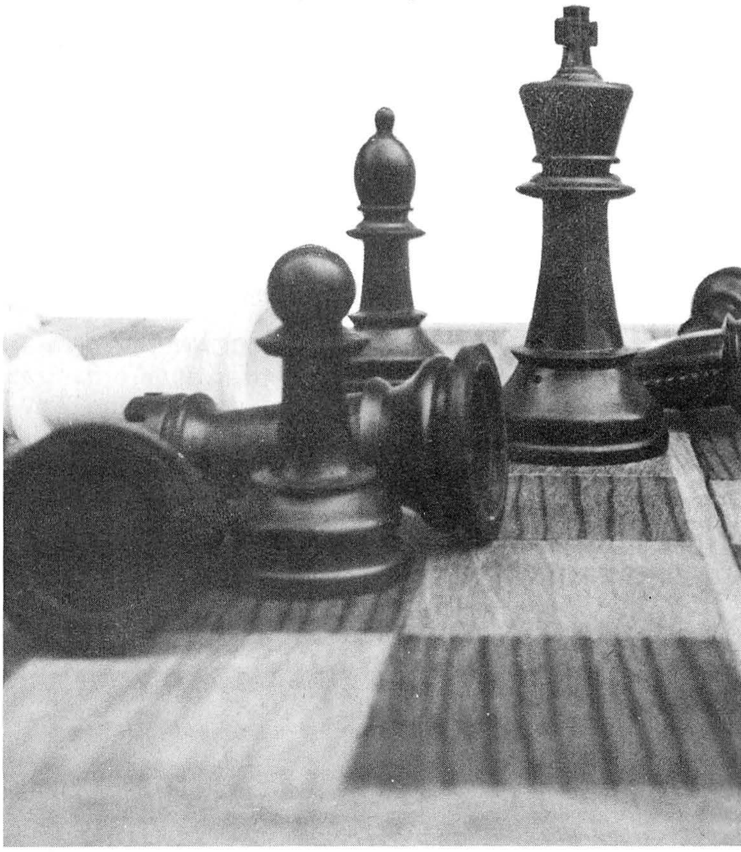
From square one, these two programs are as different as a White Knight and a Black Pawn. Microchess is quick, brash, and instruction-oriented. Sargon is ominously slow, ponderous, and strong. Since much of the difference is related to using the programs, I'll first outline the features of each.

Microchess

Microchess 2.0 runs in 16 K bytes of memory. Eight K bytes of that is a HIRES (high resolution graphics) display in black and white, with pieces that are beautifully detailed. The board is displayed next to an operations area which shows the move entry, comments, and a little twirling prompt character which shows that the program is "thinking." This is psychologically pleasing and assures you that the program hasn't crashed. Moves are entered in algebraic notation in several possible formats such as e2e4, with no hyphen required. However, only the last 2 half-moves are displayed, and each new move overwrites the old.

Microchess has an extremely useful feature: if you wish a move to be ignored (except for its effect on the display), you append an asterisk to the move. This can be used for giving odds by capturing and removing selected pieces before the start of the game, or you can correct a mistake during the game. The defect is that once a piece is gone, it cannot be resurrected. Because of this, setting up an arbitrary or adjourned position is tedious.

You can select 8 levels of play called the IQ. The manual says that at IQ = 8, the program may look up to 6 plies (1 ply = a half-move, ie: a move for one side) ahead, but this must be severely pruned in the early



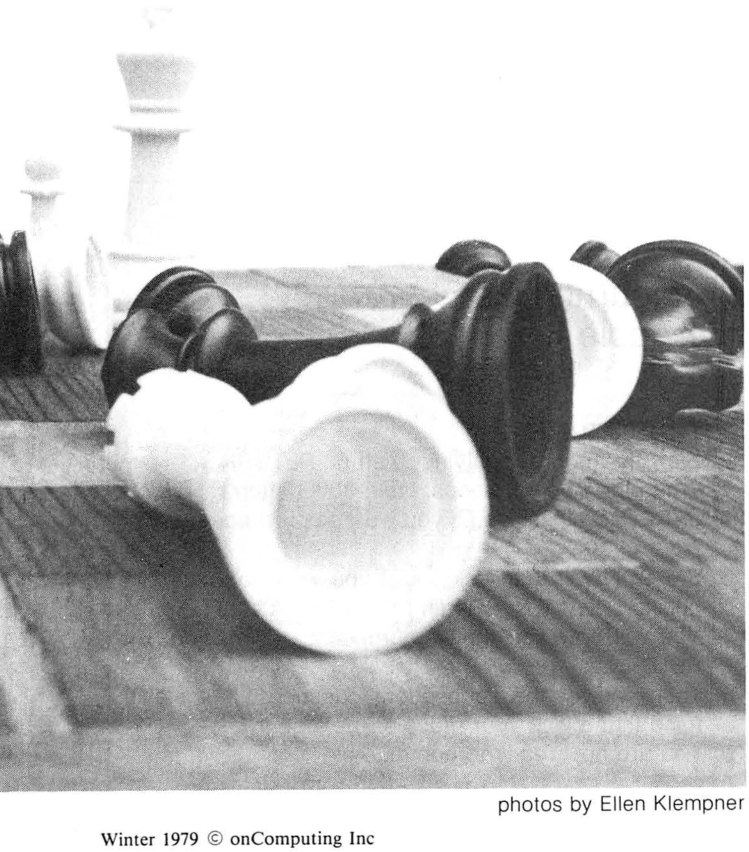
Editor's Note: This review of two chess playing programs assumes that the reader has some familiarity with the rules of the game. Either of these programs, combined with an introductory book on chess, would be an excellent tool for learning this highly addictive game.

game. At IQ level 2, a full-width search is carried out. The response never takes more than about two minutes at level 8, and is nearly instantaneous at the lower levels.

One weakness of Microchess is that in order to save memory it does not check for an illegal castling operation by the opponent. However, *en passant* moves are fully operational by both the program and the opponent.

n vs chess

Computer Chess?



photos by Ellen Klempner

Extensive use is made of the Apple's speaker. There is a double beep when the program moves (in case you were staring out the window) or when you make an illegal move, and a distinct warble when there is a check by either side. There is also a noise you have to hear to believe when the program loses.

Thoughtful instructional features are included, such as the ability to easily make the program play itself; an optional labelling of the squares for those players new to algebraic notation; and an opening book of 32 different openings to a depth of 8 moves. One thing missing is the capability to list out the entire game. Record-keeping is one thing that a computer is good at, and it would be nice to have a "list the game" function.

In summary, Microchess is nearly flawless from a user's viewpoint.

Sargon

Sargon runs in 24 K bytes of memory, with 8 K bytes used to create a high resolution display of the chessboard on the video screen. The board takes up the entire screen, and the display of the pieces is almost the same as that of Microchess, ie: excellent. By hitting the "ESC" key, you can switch to TEXT mode (and back to GRAPHICS again) for a listing of the moves made so far. Presumably then, if you have a printer, you can get a hard copy of the entire game. This is a very nice feature.

Unfortunately, there is no visible indication that the program is running, and there is very little use made of the Apple's speaker — it is beeped only when an illegal move is made. As a result, if you are not careful, you may miss the fact that Sargon has moved. In fairness, these features were not of interest to the authors, who designed the program to run on a different computer for tournament use. However, the Apple II implementation could have been smoother; more on that later.

Sargon has two operational modes: play and analyze. If you indicate that you want to play a game, it will ask you which color you want and what "look-ahead level." On the other hand, if you want to analyze a position, Sargon has a very convenient feature that allows you to create any piece on any square in preparation for such things as a 2-move mate problem or an adjourned game. Unfortunately, there is no "clear the board" function, so you have to erase the starting position piece-by-piece to set up your position.

The playing level is equal to the number of plies look-ahead, so that level 6 (the highest) is a 3-move look-

ahead. *Alpha-Beta pruning* is used to eliminate the examination of needless nodes in the tree search [Alpha-Beta pruning is a complex programming technique used to optimize the search for the right move. See "An Introduction to Computer Chess" in the October 1978 issue of BYTE magazine — ed].

Sargon is slow running on the 6502 microprocessor. The reason for this is that Sargon was originally de-

1978 West Coast Computer Faire, San Jose CA
March 3-5 1978

Place	Program	Wins	Losses	Draws	Points
1	Sargon	5	0	0	5
2	Chess Challenger	3	2	0	3
3	Commodore Chessmate	2	1	2	3
4	Boris	2	2	1	2½

Personal Computing Micro Chess Tournament
November 1978

Place	Program	Wins	Losses	Draws	Points
1	Chess Challenger (10 level)	10	0	2	11
2	Sargon (TRS-80)	6	1	5	8½
3	Boris	7	3	2	8
4	Microchess 2.0 (PET)	3	5	4	5

Results of two personal computer chess tournaments.

Procedure for Saving an Adjourned Game with Sargon on the Apple II

A. To save a position on cassette tape, wait until it is your move.

1. Hit RESET.
2. Type: 0000.5FFFW
3. Put a cassette with at least 3 minutes of space in the recorder, start recording, then hit RETURN.
4. When the Apple II beeps, you have saved the screen, the SARGON program, all its scratch pad, the HIRES display, the program pointers . . . everything.

B. To reload in adjourned game.

1. Put your tape containing adjourned game data in the recorder.
2. Type: 0000.5FFFR
3. Start the recorder playing, then hit RETURN.
4. When the Apple II beeps, type: 800G, then RETURN.
5. SARGON will ask you if you want to play a game. *Say No.*
6. SARGON will then ask you if you want to analyze a position. *Say Yes.* (Y then RETURN as usual.)
7. You will now see the adjourned board position. Hit RETURN.
8. Then answer the subsequent questions, keeping in mind that it was your move at the adjournment.
9. You are now ready to continue playing.

signed to run on a Jupiter Wave Mate computer which uses a Z80 microprocessor running at 4 megaHertz (MHz). The 6502 runs at 1.0 MHz. In a trial Ruy Lopez opening (a standard chess opening), the program at level 3 still hadn't responded with the expected 2 . . . . N-QB3 (b8-c6) after 30 minutes. At level 2 it will do so in about 2 minutes. In fact, the manual states that at level 6 in complex situations, it may take 48 *hours* to respond.

The manual advises patience.

Sargon has no opening book (ie: no prestored listing of the more popular opening move sequences). The program responds with either P-K4 or P-Q4, whichever is more appropriate. This is probably no problem when the program is running at 4 MHz, but an opening book is vital for the 6502 in order to speed up the game. The only reasonable way to play a head-to-head game with Sargon is to start at level 2 and go to level 3 only when there are fewer than 10 pieces on the board. Doing this will keep the response time down to a comfortable 2 to 3 minutes.

Sargon, in its adaption to the Apple II, has one severe "mechanical flaw": it is apparently oblivious to stalemates. For example, I positioned the pieces as follows:

Black King at	a8
White King at	d7
White Queen at	c1

Then I started the game by moving the Queen c1-c7. Sargon's response was to move to b8!! Highly illegal. Meanwhile, Microchess when confronted with the identical situation instantly and correctly replied, "stalemate."

In the listing of the game, Sargon does not log every check. It does so only if there is room on the screen. This is inconsistent and should be corrected along with the stalemate problem. One nice thing, though, is that if Sargon sees a forced mate within its look-ahead level, it will announce it.

In summary, Sargon suffers in its transformation from a program designed to win tournaments to a friendly "at home" chess opponent.

The Play of the Programs

The most objective results obtained so far are the results of the 1978 West Coast Computer Faire tournament and the *Personal Computing* Micro Chess tournament. The results indicate that Sargon is the stronger chess player. A summary of the results is shown in the box. (Not all of the contestants are listed.) In the *Personal Computing* tournament, Sargon met Microchess 2.0 twice. Sargon won with White and drew with Black.

If you forget about the mechanics of playing and concentrate on the play of the programs, they are both very good and very bad. Both programs suffer the characteristic weaknesses of microcomputer chess programs. Both are reluctant to castle at the earliest possibility (Microchess more so), and both are oblivious to the danger of passed pawns at the lower levels of play. (By the way, pawn promotion is automatic and mandatory to a Queen in these programs: there is no other option.)

Both suffer the typical "horizon effect" whereby the program seeks to delay disaster as long as possible (even if it is a mini-disaster), to the detriment of its overall position. Each program will push the pawns in front of its castled King all too easily and without sufficient provocation. Both are greedy pawn grabbers and lack positional insight. Microchess will just not mount an all-out attack, even when ahead in material and the position dictates it. These limitations are quite similar for both programs, indicating the physical limits of what can be done on a microcomputer.

The strengths of the programs are in their resiliency in tactical binds. Just when you think you have the program cornered, it will find the one move required to extricate itself (temporarily, anyway). Being a good tactician is a characteristic of these programs because mathematically that is the most straightforward approach. You must be on the constant lookout for pins, forks, and mini-combinations.

Sargon seems to be the more aggressive player, and of the two is more likely to synthesize a good defense with an attack. Microchess would rather wipe you and your *en prise* pieces off the board and hope that you fold. That is why beginners or poor chess players I've talked to marvel at Microchess. But with slow, careful play (and believe me, Sargon will give you the time) you can easily conjure up combinations beyond either program's look-ahead level. If you haven't given away any pieces outright, you can go into the end game in great shape. Here the programs suffer because they often fail to realize the potential of the pawns.

The consensus is that these two programs play at an *Elo* rating of about 1500, and I would agree with that.

A *rating* is a guide to the ability of a chess player, based on historical data of performance in over-the-board competition in rated chess tournaments. The system used most commonly was invented by Dr Arpad Elo. A player gains rating points by winning a tournament game, and loses points by losing a game. The effect of a draw on a rating depends on the difference in rating between the two players in the drawn game. A lower-rated player achieving a draw with a higher-rated player will gain points, while the higher-rated player will lose points. To provide a basis of comparison with the claimed rating of 1500 for these programs, the rating of the program Chess 4.7 is approximately 2030; the ratings of the strongest human players range from 2400

to 2800; and the rating of an average member of the United States Chess Federation is around 1300.

Conclusion

Each program has its peculiar strengths and weaknesses from a user standpoint. When it comes to the play, Sargon will take a little longer to find a slightly better move. If you are a chess novice and are anxious to get quick feedback and have some good instructional features, Microchess will do nicely. If you are more experienced and want a stronger, more cerebral program, Sargon is for you. And if you are a complete chess-computer freak, buy both and have your own tournament. (You'll need two computers.)

A final note: Sargon has been upgraded to Sargon II for tournament play. The program is currently available for Apple II and TRS-80 owners. However, we have not evaluated it as of this writing. Also, even though the TRS-80 runs at 1.77 megaHertz (MHz), Microchess 1.5 is not as strong as the improved Microchess 2.0. ■

John Martellaro is presently working towards a PhD in physics at New Mexico State University. An Apple II owner, John has long been interested in computer chess.

A Simple Cassette Directory

by John K Appleton

If your computer understands BASIC, and your cassette tape recorder has a tape counter, you can add a convenient program directory to each of your cassettes. This will allow you to have a directory at the beginning of a tape listing all the programs stored on that particular cassette.

In order to put this method into practice I will refer to my own Southwest Technical Products 6800 computer system, although this method should work on many other computer systems. My computer uses a JPC high speed cassette interface presently running at approximately 2400 bits per second. The actual data transfer rate of the cassette interface will determine the length of tape required to store a program. On my system an 8 K byte program requires about 7 feet (178 cm) of tape. The directory is a reserved area at the beginning of the tape. It will expand as more programs are added to the tape, and this expansion must be allowed for. From experience on my own system I find that allowing twice the room required to store an 8 K program gives enough room for directory expansion. Since 8 K bytes of data require about 7 feet on my system I just multiply 7 X 2 = 14 feet (356 cm), which is the amount I allow for a directory on my tapes. Therefore, the directory itself is stored starting at 001 on the tape counter (allowing 1 foot for leader) and the first program is stored starting at 015.

I used BASIC to write the cassette directory. The directory is strictly text created by BASIC and is never put into RUN mode. There are no BASIC commands used; therefore, it is not an executable BASIC program. To create your directory, type in the statement number of the line as usual, but the actual value of the line number will be equal to the starting position of that particular program on the tape footage counter. After the line number (tape footage counter), type a space followed by your own description of the program stored at that place on the tape. When you have finished typing in the directory, SAVE it at the beginning of the tape in the reserved area as mentioned earlier.

In order to access the cassette directory at any time, simply LOAD it in using BASIC, then LIST it to read the directory on the terminal. My computer's 8 K BASIC allows you to stop a listing by pressing any key on the keyboard (I always use the space bar). Then by pressing that key again the LIST will continue. This is useful if the cassette directory is larger than the screen display. I have a printer on output port #0, so by typing LIST#0 I can get a printout of the cassette directory.

When you store another program on the tape, store it just after the point where the last one finishes. When adding a program to the tape, you must update the directory using BASIC and add the pertinent information to it. The directory is then recorded at the beginning of the tape, overwriting the old directory.

Before I developed this cassette directory method I was burdened with inadequately labelled cassettes containing forgotten programs. Now my system is a pleasure to use. ■

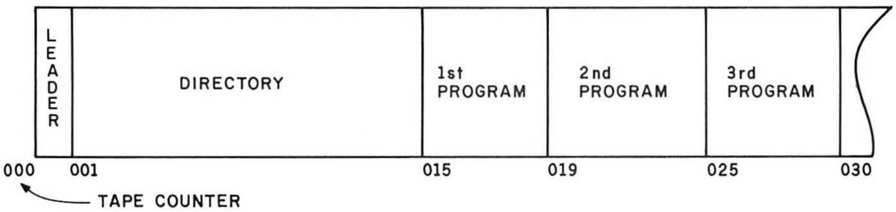


Figure 1: Physical layout for the simple cassette directory.

```
0001    -----------TAPE #6/SIDE A CASSETTE DIRECTORY---------------
0002
0003
0015    LOAN      -- LOAN COMPUTATION PGM
0019    SAVING    -- SAVINGS COMPUTATION PGM
0025    STOCK     -- STOCK COMPUTATION PGM
0030    STKGPH    -- STOCK GRAPH PGM
0036    TAX       -- TAX COMPUTATION PGM
0042    CALC      -- FINANCIAL UTILITY CALCULATOR
9996
9997    * NEXT AVAILABLE SPACE AT 049 *
9998
9999    END OF DIRECTORY
```

Listing 1: Creating the cassette directory. In order to add a cassette tape directory to your tapes, use the format shown here. The directory is actually a BASIC program. The program cannot be executed by the computer, but it can be listed by the user, which is all that is necessary. As each new program is added to the tape, simply update the directory with a new line of text, having a line number equal to the value of the tape counter at the beginning of the actual program. For instance, the LOAN program can be found at 0015 on the tape counter.

BOOK REVIEW

The Little Book of BASIC Style

by John Nevison
Addison-Wesley Publishing Co, Reading MA, 1978
151 pages, paperback $6.98
Reviewed by Noel K Julkowski

Many programs are written to work, but few are written to be read. Writing readable BASIC is the focus of this concise book of style. Since BASIC is the Esperanto of the personal computer business, this book is an excellent addition to anyone's personal computer library. Application of the principles discussed will result in better programs. Not only will your programs be simpler to write, test, and execute, but they will be easier for you and others to change.

Understanding the problem and planning the solution are the keys to efficiency, states the author. Once the plan is made, the comments are written. The purpose of the code is set down at this point, and the comments conclude with defining the variables. The code is then written. Your first attempt to run the program may fail, as mine always does. However, it is far easier to follow the logic of well-planned code and see where your mistakes are.

Mr Nevison's next point is kindness to the eye. Layout of code, including internal comments and blocks of related logic, enhances the readability of the program. Simple rules are noted which, when followed, make the meaning of the code jump out at the reader. Then when you need to change the program your job is considerably easier.

Several programs are presented to illustrate the topics discussed. They are usually 1 page long — just right for the beginner. While painlessly learning, the student can obtain some programs for the system library. Beyond the short programs lies the world of long, complicated programs. In this section the author introduces *structured programming* in a general way.[Structured programming is an orderly technique for writing efficient and easy-to-understand programs — ed.] A large collection of subroutines is displayed, implementing the rules of style.

Throughout this fine book the prose is lucid and witty. Cartoon drawings illustrate the rules of style. Not only is there a handy index, but there is also a good bibliography of reference books for the enthusiasts. Just writing a program is not enough — it should be written with "style," and this dandy little book will help you do it. ■

Microcomputer Primer

by Mitchell Waite and Michael Pardee
Howard W Sams and Co Inc
Indianapolis IN, 1976
224 pages, paperback $7.95
Reviewed by Blaise W Liffick

As you can probably tell from the copyright date of this book, it isn't exactly new. This might lead you to suspect that the book is somewhat out-of-date, and to a certain extent you'd be right. It was written before the advent of the "appliance" computer, an off-the-shelf unit you can take home, plug in, and immediately start running with prepackaged programs or begin writing your own programs in a high-level language such as BASIC.

After using an appliance microcomputer for a while, most people begin to wonder exactly what is inside of the machine and how it operates. This is where *Microcomputer Primer* comes in.

Because it was written before the big boom in appliance computers, *Microcomputer Primer* concentrates on the aspect of constructing a microcomputer from a kit. Therefore, authors Waite and Pardee go to great lengths to describe the internal structure of computers, as well as some fundamentals: what they are, where they came from, and what you can do with them. Next, they describe basic computer concepts in a general way, from the central processing unit (the microprocessor) to I/O (input/-output) interfacing. They describe the various types of memories, microprocessors, and interfaces.

Once hardware is fully explained, the authors discuss fundamental programming considerations such as registers, instruction sets, and addressing memory. Finally, the appendices cover the topics of numbering systems and classifications of memory.

Mitchell Waite and Michael Pardee obviously know their subject very well, and are able to impart their knowledge in an understandable manner. However, one word of warning: it is impossible to discuss the subjects covered in this book without resorting extensively to computer jargon. Unfortunately, the authors did not include a glossary of terms — a serious omission, in my opinion.

Therefore, if you are not comfortable using computer and electronic terminology, you are not ready for this book. If you've had some experience in the field of microcomputers and are now looking for more information about what's inside of those machines, this book is just what you are looking for. ■

The Talcott Mountain Science Center

Photos and Text by Daniel Barstow

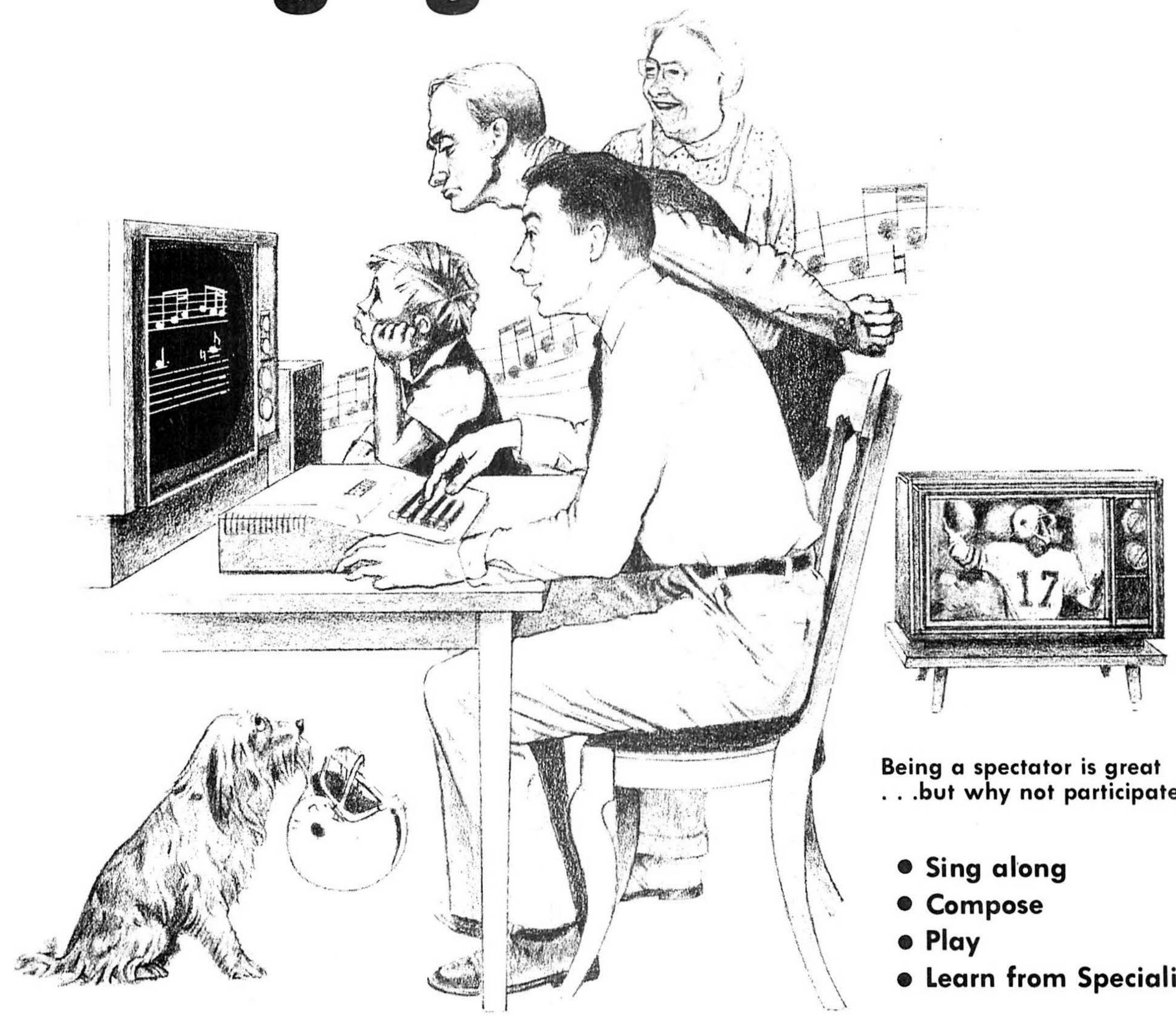
Teaching Science with Personal Computers

Every Saturday children from a number of towns throughout Connecticut travel by bus to the top of Talcott Mountain. They have been selected by their elementary and secondary schools as students with special interests or abilities in science. They are participating in a unique and exciting program at the Talcott Mountain Science Center.

Situated at the top of Talcott Mountain ridge in Avon CT, the Talcott Mountain Science Center is a regional center for science education. It has a cluster of science laboratories and classrooms, a staff of specialists, and a wealth of materials for science exploration. Students go to the Science Center to learn by doing—to become involved in real scientific investigation.

In the Saturday program for gifted and

Bringing Music Home



Being a spectator is great
. . .but why not participate?

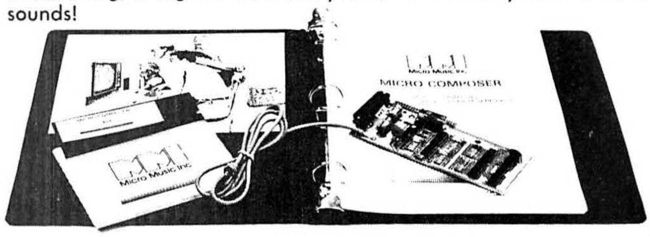
- **Sing along**
- **Compose**
- **Play**
- **Learn from Specialists**

LET MICRO MUSIC TURN YOUR APPLE II ® INTO A FAMILY MUSIC CENTER!

VISIT THE APPLE DEALER NEAREST YOU AND ASK FOR A DEMONSTRATION OF MMI'S MICRO COMPOSER™
The MICRO COMPOSER LETS YOU—

- **Play** up to **4 simultaneous voices**
- **See** all **4 voices** at the same time you're **hearing** the music—a must for music editing!
- **Enter** music notes by a fast, simple and well-tested coding system.
- **Program** the **pitch, rhythm,** and **timbre of the music. Tempo** is varied by the Apple paddle.
- **Choose 7 different tone colors** for each voice or **create your own tone color.**
- **Compose, edit, display,** and **play** music through an interactive, command-driven language that's easy to learn.
- **Save** your **music** on disk or cassette.
- **Hear** quality music sound at low cost through the **MICRO MUSIC™ DAC** card. No amplifier needed! Designed for MMI by Hal Chamberlin and Micro Technology Unlimited.
- **Select** from future MMI music instruction software to accompany the **MICRO MUSIC DAC.**

The **MICRO COMPOSER** is an **APPLE II®** compat ibile, low-cost music system designed by the folks at MMI. Our music software was designed by leading experts in music education. A simple step-by-step instruction manual leads you through entering, displaying, editing, and playing music with up to four voices—soprano, alto, tenor, and bass. You can change the sound of each voice to reed, brass, string, or organ sounds and you can even color your own music sounds!



HAVE FUN! THE MICRO COMPOSER comes complete with an instruction manual, software disk or cassette—in either Integer or Applesoft **ROM BASIC,** and the **MICRO MUSIC DAC** music card. Just plug the **MICRO MUSIC DAC** into the **APPLE** extension slot and connect the audio cable to a speaker.

Suggested retail price $220.

Ask your local dealer for information on MMI products, or contact:

MMI™
Micro Music Inc 309 W. Beaufort, University Plaza, Normal, Illinois 61761 (309) 452-6991

Circle 90 on inquiry card.

APPLE II is a trademark of Apple Computer Inc.

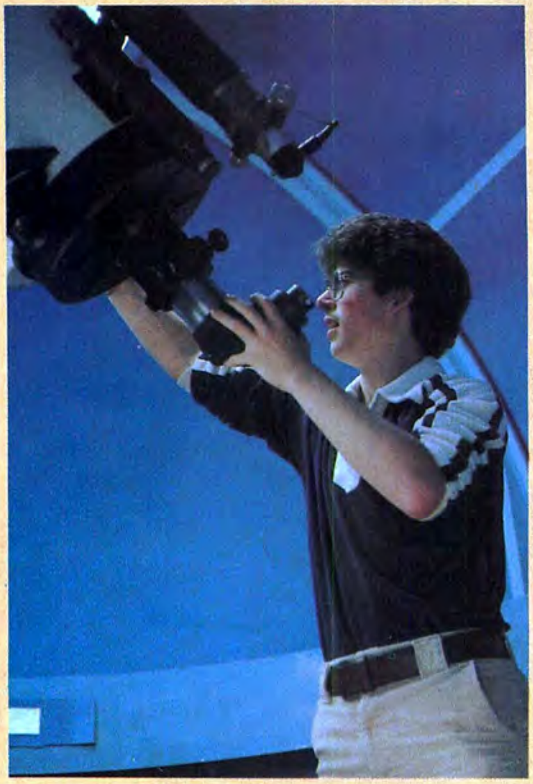


Photo 1: *Michael Dowling, a high school student at the Talcott Mountain Science Center, adjusts the center's 32 cm Cassegrain reflector telescope. He is writing a program to calculate the orbits of astronomical bodies based on observations through the telescope.*

For more advanced meteorology students, there is a program to simulate cloud seeding. Students input control factors such as rain droplet diameter and concentration, and cloud thickness.

talented children, most of the students are working on independent projects. Some students are building electronic gadgets in the radio-electronics lab, or tracking the OSCAR satellite for amateur radio communication from the ham shack. In the evening, astronomy students are using the Science Center's telescopes and developing astrophotographs in the darkroom. For meteorological research the center has a complete weather station, receives weather data by Teletype, and affords a 360° panoramic view of weather conditions throughout central Connecticut. The woods and ponds around the center provide several different ecosystems to explore. In the alternative energy lab, students experiment with solar cells, or build windmills to generate electricity. Of special interest to readers of onComputing, students interested in computers use the Science Center's PDP-11 minicomputer and Apple II microcomputer.

The PDP-11/10 (Digital Equipment Corporation) is a minicomputer capable of serving six independent users at the same time. Users can select from over 100 programs available in the program library, or can write and store their own programs. Though the mainframe of the PDP-11 is located in a corner of the Computer Sciences room, terminals can be connected from several other rooms at the center simply by plugging the terminals into special outlets. Terminals can also be connected to the PDP-11 by telephone (several schools rent computer time from the center).

The Apple II is a single-user microcomputer. It is small and portable enough to be carried from room to room, or even to schools served by the Science Center. The Apple II has a cassette tape recorder and a floppy disk drive (5 inch diskette) for auxiliary memory. The sound and color graphics capabilities are special features of the Apple. The center also has a Heathkit H11 microcomputer, and two digital logic labs. With these facilities, a wide variety of computer-related activities can take place.

Games and Simulations

Many students begin their use of the

Dan Barstow was a teacher at the Talcott Mountain Science Center. A fluent speaker of Spanish, he currently is Project Director for a bilingual gifted and talented program in Hartford. He also masquerades as Merlyn the magician, performing feats of prestidigitation.

computers by exploring the various games and simulations available on the computers. With a brief introduction to the procedures for running programs, students are soon independently (and enthusiastically) playing games and simulations such as ASTRO (a simulated lunar landing), POLLUTE (a water pollution experiment), and ANIMAL (in which the computer uses logical thought to guess which animal the user is thinking of). These programs have been selected or developed at the center specifically for their value in science education, and the students clearly enjoy their use.

Many of these programs are used in the context of other Science Center activities. Bill Danielson is Assistant Director of the center, and is in charge of the gifted and talented program. He is also the staff member with primary responsibility for the computer resources. Though his background is in astronomy and meteorology, he has been involved with the center's use of various computers since 1969. Danielson describes the use of the lunar landing program: "Before they play ASTRO, many of the students have seen our videotapes of the Apollo flights, they've observed the moon through our telescopes, they've done a "lost on the moon" activity, and they've created models of lunar craters in the classroom. Then they finish with their own lunar landing. It's a very exciting thing; they really are piloting the lunar module at that point. The visual images are important to help the computer's printout come alive."

Another instructional program is WEATHERWISE (see sample run). It was written by Danielson to help students learn how to interpret weather data. Several individuals or teams may play. Each team must pilot a ship across the Atlantic Ocean from New York to Iceland. As they trace their programs on maps, students tell the computer in which direction they would like to travel for the next four hours. The computer calculates the new locations, and also reports the local weather conditions. The students must use this information and apply their knowledge of weather maps to avoid the storms over the Atlantic. More than one ship and crew has been lost at sea over the years.

For more advanced meteorology students, there is a program to simulate cloud seeding. Students input control factors such as rain droplet diameter and concentration, and cloud thickness. The computer calculates the amount of rain produced over a given period of time. The

Photo 2: Staff member Al Vitiello demonstrates ROBAR-3, a robot he built from spare parts. The robot can be controlled directly from a personal computer.

students must try to determine the optimum conditions for cloud seeding. John Porter, a staff meteorologist, is revising the program to include the effects of updrafts and downdrafts. He also would like to use Apple high-resolution graphics to show the formation of the clouds and the falling rain.

Computer games and simulations enable students to "experiment" with scientific processes that otherwise might be impossible to experience in real life. Geology students use a program that simulates radioactive decay rates for selected materials (certainly safer than experimentation), and run a program that compresses geological time into one year. Ecology students learn about biological mutation rates with a program that simulates fruit fly (Drosophila)

Sample run of WEATHERWISE, a meteorological simulation game.

WEATHERWISE is a game/simulation to help meteorology students learn how to plot and interpret data on weather maps. Weather data is stored, or can be calculated, for any point in the northern Atlantic Ocean (over a hypothetical period of time). In practice, several teams usually play at once. Between turns, each team tears off its latest progress report and returns to the table to plot the data and determine the next bearing.

```
9WXWIS    9-JUN-79    MU  BASIC/RT-11  VO1-O1D

DO YOU WANT INSTRUCTIONS? YES
YOU ARE CAPTAIN AND CREW OF THE GOOD SHIP WEATHERWISE.
YOU MUST STEER YOUR SHIP FROM ITS STARTING POINT,
NEW YORK CITY, TO REYKJAVIK, ICELAND ACROSS THE STORMY
ATLANTIC. YOUR SHIP IS HELPED BY TAILWINDS, SLOWED BY HEADWINDS.
IT MUST AVOID SEVERE WEATHER (WINDS TOO STRONG, OR HEAVY
PRECIPITATION). EVERY 4 HOURS OF YOUR JOURNEY YOU WILL RECEIVE AN
UPDATED REPORT OF YOUR POSITION, AND THE WEATHER. THEN,
YOU STEER YOUR COURSE FOR THE NEXT 4 HOURS. THE OBJECT IS TO
ARRIVE IN ICELAND AS QUICKLY AS POSSIBLE.
ENTER DIRECTION TO STEER LIKE THIS: IF NORTHWARD, 0 OR 360.
IF EASTWARD, 90; SOUTHWARD, 180; WESTWARD, 270. IN BETWEEN,
GIVE A NUMBER IN BETWEEN. I.E.  SEE = 160.
   *  *  *  GOOD LUCK. (YOU'LL NEED IT.)



METRIC (DEG C, MB OF PRESSURE) OR ENGLISH(DEG F, INCHES OF MERC)
? METRIC
HOW MANY TEAMS ARE PLAYING? 2
ALL STARTING FROM NEW YORK?
? YES
STARTING CONDITIONS FOR ALL TEAMS:

POSITION:
LAT.  41       LONG.  73

TIME CHECK: 0  HOURS HAVE GONE BY SINCE YOU LEFT.

WEATHER REPORT:


PRESS.    TEMP.   WIND DR.   WIND SP.   CLOUDS(8THS)   PRECIP.
 1013.4   11      290        39         0              NONE.
```

reproduction. This is often done along with experiments with living fruit flies. The computer simulation can compress the months needed for several generations into the brief period of time needed to run the program.

The staff of the Talcott Mountain Science Center has also had to learn how to use these programs. The center employs about fifteen full-time science teachers. Only a few of them had any experience with computers before starting at the center. They are specialists in other fields of science, and have learned about computers on the job. Experience has shown that it is easy for teachers to learn how to use both the PDP-11 and the Apple II computers. It has also been found that the individual staff members acquire different levels of computer literacy,

```
WHAT'S YOUR TEAM NUMBER?
? 1
IN WHICH DIRECTION DO YOU CARE TO STEER YOUR SHIP?
? 155
THANKS, TEAM 1

POSITION:
LAT.  33.3        LONG.  69.1

TIME CHECK: 4  HOURS HAVE GONE BY SINCE YOU LEFT.

WEATHER REPORT:

PRESS.    TEMP.    WIND DR.    WIND SP.    CLOUDS(8THS)    PRECIP.
 1015.4   14       20          34          0               NONE.

WHAT'S YOUR TEAM NUMBER?
? 2
IN WHICH DIRECTION DO YOU CARE TO STEER YOUR SHIP?
? 180
THANKS, TEAM 2

POSITION:
LAT.  33.1        LONG.  71.4

TIME CHECK: 4  HOURS HAVE GONE BY SINCE YOU LEFT.

WEATHER REPORT:

PRESS.    TEMP.    WIND DR.    WIND SP.    CLOUDS(8THS)    PRECIP.
 1016.2   15       30          27          0               NONE.


WHAT'S YOUR TEAM NUMBER?
? 1
IN WHICH DIRECTION DO YOU CARE TO STEER YOUR SHIP?
? 90
THANKS, TEAM 1

POSITION:
LAT.  30        LONG.  65.9

TIME CHECK: 8  HOURS HAVE GONE BY SINCE YOU LEFT.
```

according to their own needs and interests. By now, all staff members at least know how to run programs appropriate to their particular fields of science. Some of the staff also know how to write programs in BASIC, and have worked on programs to serve particular instructional needs. The three staff members who do most of the teaching about computers have had more extensive experience with computers (mostly at the Science Center). In any case, there is a sharing of expertise among the staff members, and the computers are used to some extent in all areas.

Data Analysis

Several of the Science Center's instructional activities take advantage of the computers' abilities to analyze data. In one such activity, students launch helium-

```
WEATHER REPORT:

PRESS.    TEMP.   WIND DR.   WIND SP.   CLOUDS(8THS)   PRECIP.
 1011.4   15      40         51         0              NONE.



WHAT'S YOUR TEAM NUMBER?
? 1
IN WHICH DIRECTION DO YOU CARE TO STEER YOUR SHIP?
? 15
THANKS, TEAM 1

POSITION:
LAT.  61.2      LONG.  32.5

TIME CHECK: 28  HOURS HAVE GONE BY SINCE YOU LEFT.

WEATHER REPORT:

PRESS.    TEMP.   WIND DR.   WIND SP.   CLOUDS(8THS)   PRECIP.
 982.5    12      30         48         8              RAIN

HEAVY PRECIPITATION ENCOUNTERED! YOU TURN AROUND AND RETURN TO
YOUR LAST POSITION, HAVING LOST FOUR HOURS. BACK TO THE WEATHER
MAP; THEN CHOOSE ANOTHER DIRECTION.



WHAT'S YOUR TEAM NUMBER?
? 1
IN WHICH DIRECTION DO YOU CARE TO STEER YOUR SHIP?
? 30
THANKS, TEAM 1
!!!  ***  CONGRATULATIONS  ***  !!!

YOUR SHIP HAS ARRIVED IN PORT! TOTAL TIME FOR CROSSING WAS 142.36
HOURS.
UNFORTUNATELY 3 CREW MEMBERS DIED OF SCURVY AFTER YOU RAN OUT
OF ORANGES. WATCH THE STORMS MORE CAREFULLY NEXT TIME.
```

filled balloons to determine wind speed and direction. They use a theodolite and stopwatch to accurately measure the changes in balloon positions over time. The students run a special program on the PDP-11 to analyze this data and calculate the observed wind conditions. One group of students used balloon launch data from several locations to study pollution dangers associated with a proposed interstate highway. The study became one of the factors used in a decision to relocate the highway away from a group of water reservoirs.

The center has solar prominence and sunspot telescopes, as well as a solar spectrometer. Various programs are used to analyze sunspot activity and calculate solar temperatures. Programs like these help students interpret and understand research data.

Chronobiology is the study of biological processes as they rhythmically change with time. For example, temperature in human bodies varies according to regular patterns throughout the day; plants blossom in annual rhythms; the human menstrual cycle is approximately twenty-eight days. (Chronobiology should not be confused with the pseudoscience of "biorhythms.") Science Center students use a range of equipment to study rhythms in plants, animals, and humans. The computer is an essential tool for this study.

Dr Donald P LaSalle, Director of the Talcott Mountain Science Center, has been a pioneer in educating children to use biological measurements to learn about their own circadian (daily) rhythms. LaSalle comments, "The computer is to chronobiology as the telescope was to astronomy. Galileo's telescope enabled him to see detail precise enough to recognize the true nature of the solar system. The computer enables us to study large enough amounts of biological data to recognize subtle changes. Before computers, sophisticated chronobiological research was virtually impossible. With computers, even elementary school children can take their own biological measurements and run the data analysis programs." This information

Photo 3: (left) The center's PDP-11 minicomputer, which can be connected to six terminals throughout the complex.

Photo 4: (above) Children playing WEATHERWISE at the computer terminal.

can help students recognize their own daily rhythms, and understand how these rhythms affect their daily lives.

Student Programs

After a few sessions working with the computers, most students are eager to learn how to write their own programs. In the Saturday program, students may choose to participate in a six to twelve week session on computer programming. The emphasis is on applying the computers to the students' work at the center.

BASIC is the programming language used for both the PDP-11 and the Apple II (some of the more advanced students opt to learn machine language as well). Though BASIC has some limitations, students find it easy to learn and apply.

Photo 5: Adjusting a solar heater to boil water. Based on the recorded temperature data, another student wrote a program to calculate the amount of time necessary to raise the oven to any selected temperature.

The staff has found it helpful to use a "read-modify-write" sequence to teach particular commands. In the case of FOR-NEXT loops [a loop is a type of computer language structure in which the computer repeatedly cycles through a series of instructions], for example, students first read a simple program which contains a FOR-NEXT loop, while the teacher demonstrates its use. Next, they observe the effects of modifications in the program, such as moving the NEXT statement. Finally, the students write their own programs using FOR-NEXT loops. This sequence can be applied to any command, and helps students progress from the concrete to the abstract.

Some early programs include: mathematical calculations such as area and

The computer is a powerful, multipurpose tool that expands the intellectual reach of both students and teachers.

average, graphic manipulations, word quizzes, a coin toss simulation, and conversions such as Fahrenheit to Celsius. It takes just a few weeks of practice for students to feel comfortable writing programs in BASIC.

As students begin to work on longer programs, the staff has found it important to teach structured programming techniques. (Structured programming refers to a particular approach, and techniques, for writing programs that are logically organized and clear to understand—human readability is as important as computer readability.) Students are shown how to use a "top-down design," which involves breaking the overall program design into smaller, more manageable sections. These sections can serve as the basis for "block-structured" programs. REMarks are used to indicate the flow of control in the programs, and explain the operations of sections that might be confusing or unclear. A heading is required on all student programs to indicate the name of the program, the author, and a "save until" date (student programs are periodically purged). Preferably the heading also includes a list of variables and a brief description of what the program does. There are some limitations on the use of "true" structured programming techniques, because BASIC lacks certain control structures such as WHILE and UNTIL loops. Nevertheless, the staff has found it essential to specifically teach a logical approach to program design. Otherwise student programming projects become confusing and unmanageable.

Many of the students' programs are related to other areas of interest. A student who has had some experience flying airplanes is presently writing a game program which involves a simulation of the airplane's controls. Various different scenarios can take place, and a successful flight requires careful calculation of flight

direction, wind speed, fuel consumption, and altitude.

An ecology student, who had been studying about the predator-prey relationship of rabbits and foxes, recently made a computerized model of a forest. The program calculated the total populations of rabbits and foxes, based on such factors as initial populations, food available in the forest, rainfall, and natural disasters. Defining these interrelationships in mathematical terms became the focal point for a careful ecological analysis.

Lisa Barnhart, a high school student, used the computer to study the correlation between crime and weather. She based her study on crime statistics from the Hartford police, the state police, and the FBI, along with weather data for a five year period. Quoting from her summary report: "The temperature had the greatest influence on crime. It had its largest effect on larceny, then in descending order, on aggravated assault, burglary, and robbery... Precipitation, strangely enough, had a great effect on two crimes: robbery and burglary. I had anticipated a lower crime rate when there was more precipitation."

Michael Dowling, another Science Center student, is working on a program which will calculate the orbits of planets, comets, asteroids, and other orbiting objects, using the Laplacian method. The calculations are based on position data from three precise observations through the center's 32-centimeter reflector telescope (see photo 1).

A geology student wrote a program to identify rocks based on descriptive data. By asking questions about rock characteristics, the program is able to identify several types of rocks. The classification scheme underwent several revisions as the student learned more about rock classification systems.

The color graphics of the Apple II offer powerful capabilities that the students enjoy exploring (see photo 6). In some cases the students experiment with kinetic art, manipulating the graphic images under program control (see listing 1). One group of students used the Apple to display a graphic image of the sun and indicate the placement of sunspots according to daily observations. Their program attempted to duplicate the correct motion of the sunspots.

A project not yet completed is the storage of star position data in the Apple II. Users will be able to specify particular areas of the sky, and the Apple II will use high-resolution graphics to plot the appropriate stars in their correct positions. When it is completed, the Apple II can be easily carried to the center's observatory to facilitate stargazing.

Though some of these programs may seem fairly sophisticated, few of the students had any experience with computers before going to the Talcott Mountain Science Center. Once students are familiar with a programming language such as BASIC, writing programs is not a difficult process. The focus of their work usually is on designing the algorithms, or procedures, which form the basis for the program. Thus the emphasis is primarily on scientific understanding of the problem. Actually writing the program (and implementing, testing, and debugging) involves time and work, but is rarely too difficult for the student to handle.

The major problem with computer use at the Science Center is providing sufficient computer access time to the students. In spite of the availability of six PDP-11 terminals, the Apple II, and a few other computer resources, demand for computer time is higher than the supply. There are several aspects to the solution, none of which completely alleviate the problem. The computer programming class has a lower student capacity than some of the other classes—twenty students (with two teachers) is a maximum. With the Apple, and four or five terminals to the PDP-11

Photo 6: Students learning how to use the color graphics of the Apple.

(some terminals are usually used by other classes), this provides a student/computer ratio of about four to one. Sometimes students work in groups, and there is a sign-up procedure to take turns. The center also uses student tutors to help answer questions, both between turns and at the terminals. It's important to remember that the off-line time is also vital to the program development process. Students need to study the problem and design the program before sitting at the terminal. Also, corrections and improvements can often be seen more clearly when the programmer gets away from the immediacy of the keyboard and printout.

Hardware Projects

"We definitely are software oriented," reports Danielson. "It's clearly easier for students to write their own programs than to design hardware construction projects. However, a number of our students have gotten involved with hardware design and use our electronics lab to build computing machines."

The Science Center has a few commercial digital logic labs, which help introduce students to the theory and practice of digital design. The center's first microcomputer was a Heathkit H11, a kit built by a group of students. Other students have built computer kits for

Listing 1: The random function is popular with children to add an aspect of chance to their programs. 9-year-old Robert Jellinghaus wrote this program to draw random points and lines, of random colors, using the Apple's color graphics. Note the feature to stop the changing display when the user likes a pattern.

```
10    REM RANDOMPLOT
20    REM BY ROBERT JELLINGHAUS
30    REM SAVE UNTIL DECEMBER 31, 1979
40    REM DRAWS RANDOM PICTURES ON AN APPLE COMPUTER
50    DIM A(3), B(8)
60    REM   VARIABLES
70    REM     A( )        RANDOM COLORS
80    REM     B( )        RANDOM POSITIONS
90    REM     C1          COUNTER VARIABLE
100   REM     C2          COUNTER VARIABLE
110   GR
120   PRINT "TYPE CTRL/C WHEN YOU LIKE THE PATTERN"
125   PRINT "TYPE CONT TO CONTINUE";
130   REM
140   REM SELECT 3 RANDOM COLORS
150   FOR C1=1 TO 3
160   A(C1)=INT(RND(0)*16)
170   NEXT C1
180   REM
190   REM SELECT 8 RANDOM POSITIONS
200   FOR C2=1 TO 8
210   B(C2)=INT(RND(1)*40)
220   NEXT C2
230   REM
240   REM PLOT POINTS AND LINES
250   COLOR=A(1)
260   HLIN B(1), B(2) AT B(3)
270   COLOR=A(2)
280   VLIN B(4), B(5) AT B(6)
290   COLOR=A(3)
300   PLOT B(7), B(8)
310   REM
320   GOTO 140: REM REPEAT UNTIL CTRL/C
330   END
```

personal use. With the help of Bob Judd and Al Vitiello, the center's electronics specialists, some students have designed and built their own computers.

Tom Birdsall is a high school student from Andover CT. He describes his experiences with "homebrew" computers: "Two and a half years ago I felt that I wanted to learn how computers worked by building a microcomputer—not from a kit, but from scratch. My computer is based on an 8080 microprocessor, and has 1 kilobyte of programmable memory. The first version was built on a protoboard. My second version features a case and a switch register for input/output. Though it is currently limited to mathematical manipulations, in the future I hope to use my computer to synthesize music, control remote equipment, and, above all, as a general-purpose home computer."

The Saturday program is certainly the most intensive experience students have with computers at the Science Center. Yet it should not be thought that only the "gifted and talented" can learn how to use computers. Throughout the week the center offers numerous other computer-related services to educational institutions at all levels. (It is important to keep in mind that the computer is just one of many instructional and investigative tools used at the Science Center.)

Computer Literacy

Some schools contract the center to teach basic computer literacy, either in the schools or at the center. The purpose is to expose students to the capabilities of computers, and to provide some experience in their use. The minimum is a three session sequence, which includes an introduction to computers, a chance to play some computer games and simulations, and some introductory experiences with BASIC. Several schools rent computer time from the center, with the benefit of access to its wealth of science-related programs. The Board of Education in Southington CT recently purchased an Apple computer, and has contracted the services of the Science Center to help them learn how to use it in their schools. Staff member John Porter travels twice a week to Southington to teach students and teachers how to use the Apple and how to program in BASIC. He also offers suggestions on how to apply the computer, especially in science education. Porter enjoys using the computers, and his enthusiasm is an important factor in turning kids on to computers.

Research

For several years the center has been involved with research on solar energy. Two of the buildings at the center have solar panels providing some of their energy needs. The new "Solar-heated Chronobiology and Appropriate Technology Laboratory" has several banks of active solar collectors on its roof. They provide all of the hot water and 50% of the space heating energy resources. In addition to collecting information about the efficiency of these panels, the center also operates solar instruments that measure direct and diffuse sunlight throughout each day. The data from these measurements is being collected and analyzed by computer to help determine the potential for solar energy in this area.

"The basic philosophy of the Science Center is to involve students in scientific exploration. We have found the computer to be an essential tool in this exploration."

Other agencies have contracted the use of the Science Center's computer for analysis of research data. The American Radio Relay League conducted a study of its members (amateur radio operators) to determine their interests, activities, and purchases related to ham radio. Over 3000 responses to the questionnaire were received, and analyzed at the center by the PDP-11. The PDP-11 is used for most of the research because of the large capacity of its two hard disk drives (2.4 megabytes of information per disk).

Administrative Applications

There also are administrative uses for the computers. The mailing list for the center's news bulletins has been computerized. This certainly speeds up

preparation of address labels. It also enables selected notices to be sent to specific people, such as elementary school principals or science department chairpersons. The computer is also being used to keep track of the center's inventory of materials. Danielson is working on a staff scheduling module, which should alleviate some of the problems involved in the complicated process of matching school needs, staff abilities, and Science Center resources. These administrative applications are run for the most part at times when students are not using the computers.

The Future

What about the future? The Talcott Mountain Science Center is always exploring new areas of science education. Because personal computers are becoming more popular, the center is likely to offer evening classes on the use of microcomputers and how to program in BASIC. There may also be classes in digital logic for those interested in hardware experimentation.

Since many public schools are also buying microcomputers, the center will be providing special training courses for teachers to learn how to use computers. This will include demonstrations of the computer's applications in science education. Most of the science-related programs developed at the center can be used in other schools, even if they do not have the extensive resources available at the center.

Some hardware acquisitions are likely. A light pen, bit pad, or other graphic input device would facilitate the use of Apple graphics. The Apple's existing 16 kilobyte capacity of programmable memory is inadequate for some applications, so another memory card will probably be purchased to expand capacity to 32 kilobytes. Since a number of the center's instruments are analog devices, analog-to-digital (A/D) converters are needed to interface these instruments with either the PDP-11 or the Apple II. (Analog refers to continuous change, such as a moving second hand on a clock. Digital refers to counting by specific numerical increments, such as the changing time in digital watch displays. Most computers can process data only in digital form.) The center may also explore the use of speech generation and speech recognition devices.

The most important aspect of improving the educational uses of the computers is new software. Some programs will be purchased, especially since the boom in microcomputer use is motivating software companies to make available more programs. However, it has been the Science Center's experience that programs applicable to its needs are best developed by the staff.■

Bob Judd, head of the ham radio station, is writing a program to use the Apple II's speaker for a Morse code drill. Another staff member is writing a program to display various electronic circuits and a chart of calculated values at different points in the circuits. This is in response to some student confusion related to voltage and current in circuits. Staff members are also working on a program to plot data from weather stations on a computer-generated map of the United States (based on map generation programs in *BYTE* magazine, May and June 1979). The computer has even been programmed to perform magic, as part of an evening course on "Science and Magic" [See "Magic for Your Micro" in the Fall 1979 onComputing...ed]

The computer is definitely having an impact on science education. At the Talcott Mountain Science Center, computers have been used for ten years, and have become an integral part of the way science is taught. The computer is a powerful, multi-purpose tool that expands the intellectual reach of both students and teachers. To quote Dr LaSalle, "The basic philosophy of the Science Center is to involve students in scientific exploration. We have found the computer to be an essential tool in this exploration."■

A VOICE FOR BILL

Bill Rush rolled up to the front desk and began to spell out his request on the language board mounted on his motorized wheelchair. "I would like a Selleck Quad T-shirt please. The money is in the bag on the back of my chair." I happened to be walking by the desk and paused to see what the desk worker's reaction would be. After a few moments and several false starts, the interchange was underway. I moved closer to the desk to get a better view of Bill's spelling board. He spelled, "I bet you didn't notice that the name 'Selleck Quad' has a double meaning." (Bill and I live in the Selleck Quadrangle dormitory at the University of Nebraska.)

At this point, I became very interested. "What's the other meaning, Bill?"

He looked up and smiled. "Quad means Quadrangle, but it also means Quadriplegic."

Photo courtesy of UNL-Inst. of Ag. and Nat'l Resources

Bill Rush has lived without speech since birth. Now, thanks to a personal computer, he can "talk" with his own "Bionic Voice"

Photos and Text by Mark Dahmke

The word "Quadriplegic" was hard for me to follow. "Wait a minute — let me get a pencil. Now spell it again Bill." "Q-U-A-D-R-I-P-L-E-G-I-C" he continued spelling. "I am the other Selleck Quad." "Oh yes, of course," I responded, taking a moment to catch the pun.

In spite of the fact that Bill has cerebral palsy with quadriplegia and no speech, he is the same as you or me: just a guy going to college. He is a journalism major and a published author. My first encounters with Bill were in Selleck, the dorm where we both live for nine months out of the year.

I would occasionally meet him coming out the door and would hold it open, or stop to help him if his wheelchair got stuck somewhere. It seems that there are always people ready to assist him — out of common courtesy, never pity.

Bill uses a headstick (a long stick attached to a headband) and a language board to communicate. He also uses a Selectric typewriter to do his homework and write articles. He has a set of environmental controls in his room which allow him to turn many electrical devices on and off by remote control. Selleck Quadrangle, as well as most buildings on campus, is set up to be accessible to persons in wheelchairs, so he does not have much trouble getting around.

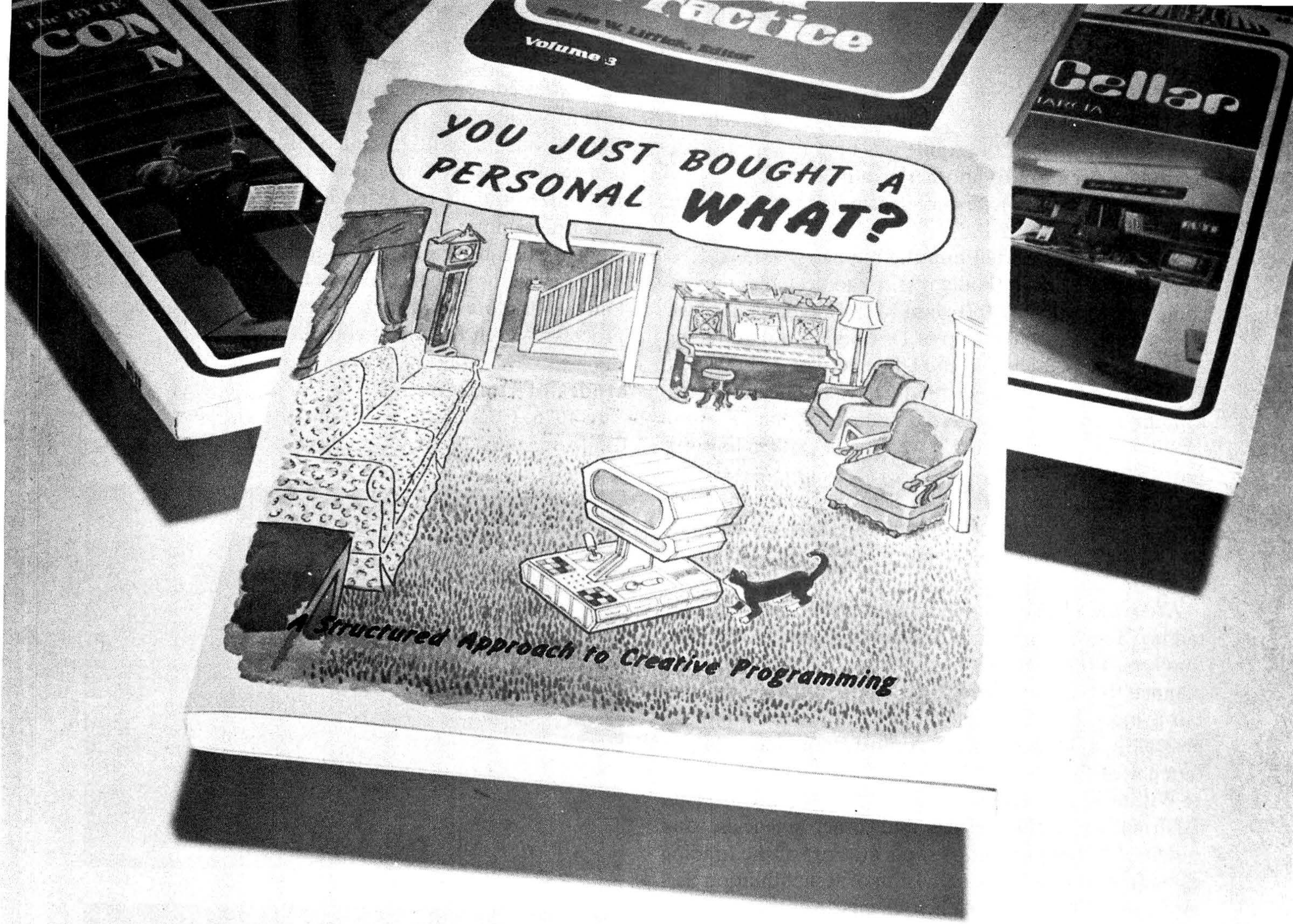
Yet there are still many barriers in Bill's way, communication being the primary one. Although he can type and use the language board, it is not the same as talking. Bill once asked, "Ever try to spell things in a dimly lit bar?"

The secondary problem is speed. Bill types about as fast as a single-finger, hunt-and-peck typist. At best this means one or two characters per second, so even a simple "Hello" can take an entire minute. Every word (except those on his board) must be meticulously spelled out. Anything complicated can take five minutes to spell and be understood. Unfortunately, most people have trouble retaining a whole phrase or sentence. By the time the end of the sentence is reached, the first part is forgotten in the effort to concentrate on the individual words.

The World of Computers Invades

The first time that I talked with Bill was at a conference on computing at the University. He became interested in computers while taking a FORTRAN course as an elective. I found him investigating an Apple II that was on display. He was trying to ask the owner if the Apple had a shift-lock key. To most of us, a shift-lock key is not a very noteworthy or important feature, but for Bill it is a necessity.

I stopped to say hello. "Bill, we haven't met, but I live in Selleck and I've seen you around a lot." He nod-



You Just Bought a Personal What?

by Thomas Dwyer and Margot Critchfield

Whether you are a novice programmer or an experienced computer user, this book is filled with practical ideas for using a personal computer at home or work. It will take you through the steps necessary to write your own computer programs as well as offering over 50 ready-to-use programs written in Radio Shack TRS-80 Level II BASIC.

You Just Bought A Personal What? will be available at your favorite computer bookstore or direct from BYTE® in February 1980. Price **$11.95**. ISBN 0-07-018492-5

BYTE BOOKS™

"BOOKS OF INTEREST TO COMPUTER PEOPLE"

70 Main Street
Peterborough NH 03458

Circle 20 on inquiry card.

ded, then began to spell.

"Does the Apple have a shift-lock?" He asked me because the owner of the Apple was busy demonstrating a graphics program to another group of people.

I replied, "No, but we could easily wire one in if you want to get an Apple."

Then I decided to tell him the big news.

"Bill, I'm glad we finally met. I had been planning to come talk to you. I didn't want to say anything until we were sure, but with the help of Dr Lois Schwab (director of Independent Living at the University of Nebraska at Lincoln), we think that we finally have financial support to build a voice synthesizer computer for you."

Bill became very excited. I went on to explain how the project had started.

Back in September I met Dr Schwab, almost by accident. I work in Academic Computing Services on campus. We went out to her department to discuss a grant proposal for academic computing.

After the initial grant business, we found ourselves talking about computers, prosthetics, and voice synthesizers. I told her that back in high school I had once planned to build a simple voice synthesizer for a science fair project. Dr Schwab suggested that we look into the possibility of a pilot project. Almost simultaneously, we thought of Bill.

Within hours, Dr Schwab contacted the University Affirmative Action office and other agencies, and obtained a commitment of support. Part of the funding came from the Nebraska Division of Rehabilitation Services, since Bill is one of their clients. Another part will come from United Cerebral Palsy of Nebraska, and the rest from the University Affirmative Action office.

Dr Schwab has been interested in microprocessor applications for the severely disabled since 1975, when she began negotiations with the Vocal Interface Division of the Federal Screw Works of Michigan (the leading manufacturer of voice synthesis hardware at that time). They had just released an $11,000 communications machine. Negotiations fell through because Votrax said it would take $250,000 and two years to complete the system she had in mind.

Designing the System

Everything happened so rapidly that I had not had time to think. Suddenly, someone was funding the project I had dreamed about since 1973. It was never completed then, because of the expense of the hardware. Things would be easier now that someone else was paying for the synthesizer, and I already owned an 8080-based computer that could be used to write the software.

After purchasing the hardware and estimating the development costs, we discovered that we could put the proposed system together for around $3,000.

Developing the dictionary handler program took up most of my free time for several months. The result was 2500 lines of assembler language that comprise a dictionary based vocabulary management system (VMS).

The vocabulary management system has many features built into it. I wanted Bill to have something that would be sophisticated and capable of being customized and altered as needed, without having to call in a programmer to modify it. I wrote it in assembler language because BASIC is far too slow for this application. Searching a dictionary that could eventually contain 500 or more words and phrases is a time-consuming process — even for a fast computer.

Modes of Operation

In 1976, Computalker Consultants of Santa Monica, California introduced a speech synthesizer for S-100 bus

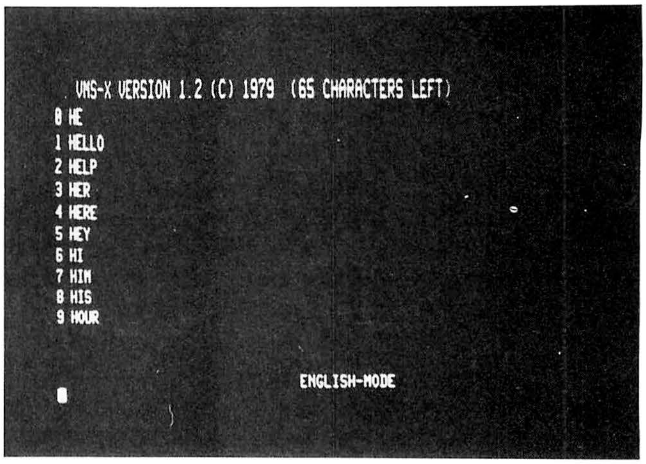


Photo 2

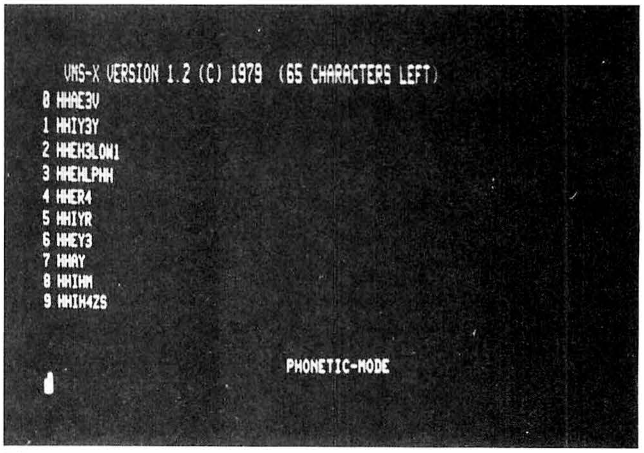


Photo 3

computers. The Computalker hardware consists of a single S-100 circuit board containing a series of oscillators and filters that can be directed to generate sounds that are produced by the human larynx and mouth. The board is an electronic simulation of the mechanics of human speech. A series of output ports (ie: locations on the computer enabling it to exchange information with the outside world) are used to send commands to the individual circuits on the board to change the voice pitch, amplitude, resonance, aspiration, and nasal quality.

The board currently retails for approximately $400,

but is similar in performance to the considerably more expensive ($11,000) Votrax speech synthesizer. Computalker also provides software support for the synthesizer, including parameter editing programs (for direct manipulation of the control voltages), and a synthesis by rule program called CSR1.

CSR1 allows the user to type in groups of phonemes (the building blocks of spoken words) to make up words, phrases, and sentences (see text box). Most of the time Bill will want to type in English words and not bother with the phonetic spellings required by the Computalker CSR1 program. The purpose of the dictionary of words is to provide a way to store and find phonetic spellings.

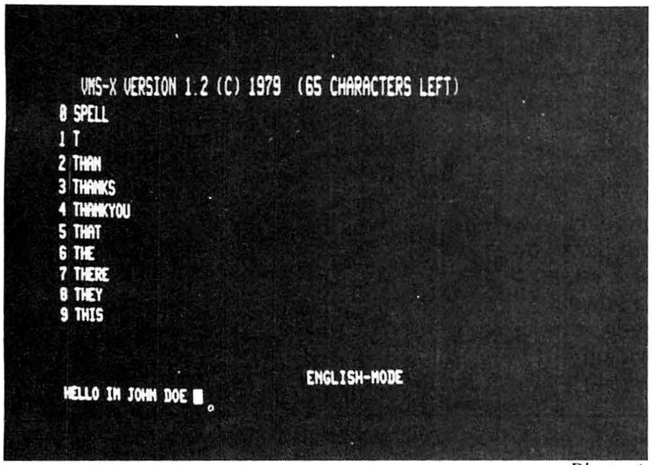


Photo 4

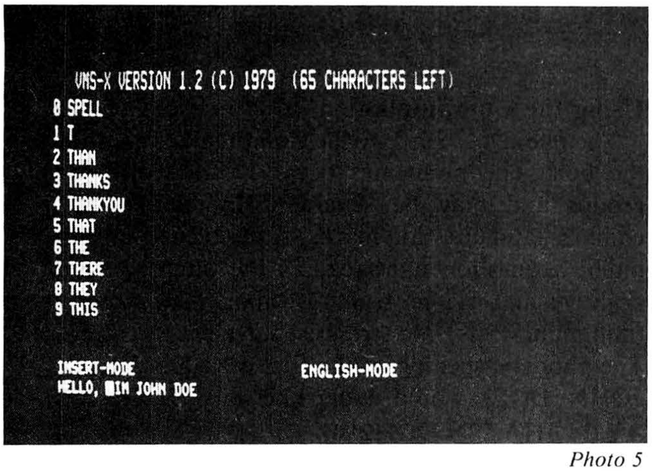


Photo 5

Photo 2 shows the video display as it will be set up for Bill. The top line is a title and version number stamp — mainly for version identification. The number to the right indicates the number of characters or bytes of storage left in the dictionary space in memory.

The lines numbered zero to nine are display lines that act as a "window" into the dictionary.

The next two lines are blank in the photograph, but are used for error messages, prompt messages, and so on. The next line is blank, except for the words "ENGLISH MODE". This field is updated whenever the mode is changed from ENGLISH to PHONETIC or NUMBER-LETTER.

Photo 3 shows what happens when the mode is switched to PHONETIC. The ten lines of dictionary display now show the phonetic portion of each dictionary entry. Line 0 contains the phonetic spelling of HAVE, line 1 contains the word HE, and so on. Note that line 1 in photo 3 corresponds to line 0 in photo 2.

Photo 4 shows a sample sentence typed in the text entry area of the screen (the bottom two lines). Text will be automatically "wrapped around" to the next line. Up to 128 characters may be entered at one time in this way.

When a sentence such as "HELLO IM JOHN DOE" is entered, and RETURN is depressed, the program will separate the words, search the dictionary for each word, replace the English word with its phonetic equivalent, call the Computalker program "CSR1," and vocalize the sentence in one continuous stream. All punctuation including periods, commas, question marks, and spaces will be left as is in the sentence. Suppose, in the above example, I discovered, after typing in the sentence, that I really wanted a comma after the word "HELLO." The program emulates (simulates) an intelligent terminal, with some very sophisticated editing features. The text may be corrected before it ever gets to the search and decode stage. In photo 5 a comma has been inserted using the insert mode function. In the insert mode, characters are inserted, and existing characters are moved over from wherever the cursor (white box) is.

In the next example, the phrase "THE QUICK BROWN FOX" has been entered (photo 6). When RETURN is depressed, the program begins to search the dictionary for each of the words. Photo 7 shows the response: "WORD "BROWN" NOT FOUND IN DICTIONARY." The vocabulary management system then prompts the user to give a response. If a phonetic spelling is entered, the response will be added to the in-memory dictionary, along with the English spelling of the word, and the sentence will be completed and vocalized. If the user decides not to enter the phonetic spelling and merely hits RETURN, the sentence will be vocalized and the missing word spelled out letter by letter.

As mentioned before, the keyboard has a group of keys that have English words engraved on the faces (see photo 8). These keys work like regular keys in both the ENGLISH and PHONETIC modes by entering the whole word as if the user had manually entered it. In the NUMBER-LETTER MODE, depressing a key will cause the single word or phrase assigned to the key to be instantly vocalized. The letters of the alphabet and the numerals will also be vocalized when hit. This mode permits the synthesizer to be used as a spelling board.

The DIRECT-EXEC-MODE works in a similar way, but causes the number keys on the keyboard to respond as lines on the screen. Thus, if '0' is depressed, the word or phrase on line 0 on the screen would be spoken.

Phonemes Used by the Computalker
Voice Synthesis Program CSR1

CONSONANTS

Phoneme	Usage
P	pie
T	tie
K	key
B	by
D	die
G	guy
M	my
N	nigh
NX	hang
F	fie
V	vie
TH	thigh
DH	thy
S	sigh
Z	zoo
SH	shy
ZH	vision
L	lie
W	we
R	rye
Y	you
HH	high
CH	chime
JH	joy
WH	why
EL	battle
EM	bottom
EN	button
Q	(glottal stop)

VOWELS

Phoneme	Usage
IY	heed
IH	hid
EY	day
EH	head
AE	had
AA	pod
AO	pawed
OW	hoed
UH	hood
UW	you
ER	herd
AH	bud
AY	hide
AW	how
OY	boy
AX	about
IX	David
OH	core
UX	too
KX	coo (K before back vowel)
GX	goo (G before back vowel)
RX	card (R after a vowel)
LX	kill (L after a vowel)
DX	pity (T between vowels)
YX	diphthong ending
WX	diphthong ending

Punctuation

space	word boundry
,	pause/silence
.	falling pitch
?	rising pitch

Stress Marks

0	no stress
1	maximum stress
2	secondary stress
3	tertiary stress
4	etc.
5	no stress (= 0)

Dictionary Storage

My original design called for an 8080-based computer with 32 K bytes of programmable memory and a digital cassette deck running at 720 characters (bytes) per second. I later found that the cassette deck I wanted would not be available for twelve weeks. I did find a used North Star disk drive and controller that a friend was willing to sell for about the same price as the cassette drive. Since I do development work on my own ICOM minifloppies, I was able to write the custom software necessary to make the North Star disk I/O (input/output) subroutines look like the ICOM routines.

The disk interface is used for two purposes in the system: to load and start the vocabulary management system and for large-scale dictionary storage. When the system is turned on, a dictionary file containing about two hundred words and phrases is automatically loaded into the memory workspace set aside for the dictionary. Later, the user may wish to load a specialized or customized dictionary. Since it would be wasteful to redundantly store the most frequently used words in each custom dictionary file, the desired file is simply merged with the existing in-memory dictionary. The merge is done alphabetically so that all new dictionary entries are brought in and put in order. The other kind of file load is called "APPEND." It works the same as a load, but first checks incoming words and phrases against the in-memory dictionary to see if they are already there. If found, the entry coming in from disk is ignored. I plan to add more features to the disk interface at a later date.

Using the Computalker

The phonemes used by the Computalker are shown in the box. The phonemes occur in one- or two-letter groups and may be separated by blanks, periods, commas, and question marks. A period causes a drop in pitch, a question mark causes the pitch to be raised slightly, and range from 1 (maximum stress) to 5 (minimum stress). No stress is equivalent to a stress of 0. The phonetic spelling of the first example (HELLO IM JOHN DOE) would look like this: HHIX3LOW1, AY3M JHAA2N DOH2W.

After installing the Computalker synthesizer and loading the CSR1 program, I began to experiment with the phonetic spellings in an effort to learn how to make words come out intelligibly. After about an hour of playing I mastered most of the frequently used combinations and rarely had to refer to the list. On the average, I find that I have a success rate of about 75 percent on the first try at typing in a word phonetically. With regard to speed, about one second of computer time is required to process and set up three seconds of speech, prior to vocalization. The speed of voice output is not affected by the speed at which the text was typed in. All words come out at a consistent speed—that of normal speech. Once a word, phrase, or sentence is

stored in the dictionary, recalling it may take only two or three keystrokes, further improving the response time.

The Computalker hardware is capable of excellent voice reproduction, although the CSR1 program synthesizes speech with algorithmic techniques and has difficulties with some sounds and combinations of sounds. I have been working closely with Computalker Consultants to improve the voice quality. One advantage of the Computalker is that the phonetic translation is done in software rather than hardware. If a new and better version of the CSR1 program comes out, the old version can be replaced at one tenth the cost of new hardware.

Other Capabilities

The synthesizer can work with a variety of input and output devices. This is because every potential user will have different needs and capabilities. The user could design a keyboard with only predefined words and phrases on it, one with just standard alphanumeric keys, or one with any other combination of keys. The keyboard designed for Bill has a standard alphanumeric layout plus a group of thirty-four word keys and a set of command keys. If the system were to be used by a person with poor coordination, a keyboard with one inch square keys could be designed and plugged in. The possible variations are unlimited.

The other advantage of the interchangeable keyboard is upward compatibility. If someone buys a system and later decides that more features are desired, a change of keyboard would make the extra features available without having to replace the whole system. If the user of the system is a young child, the first keyboard may have only a few predefined phrases and words. When it is outgrown, the system does not become obsolete; the keyboard does. The child can learn how to use it a *little* at a time. Also, the labels on the keys do not have to be in English. Foreign languages, symbolic codes, or even pictures could be used.

The computer that runs the system may be used for other purposes besides voice synthesis. A variety of software is available for the 8080 microprocessor/North

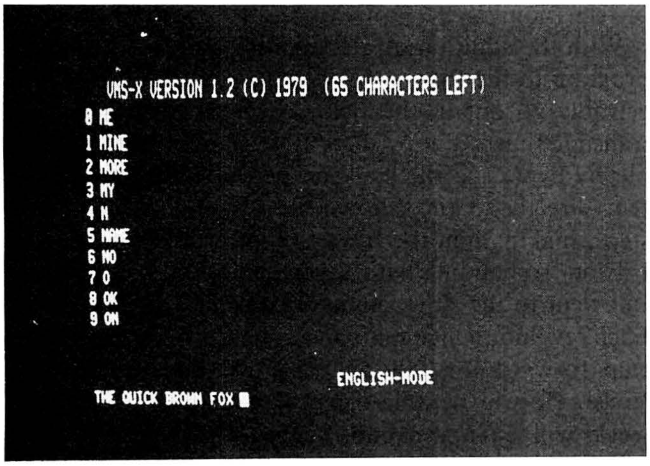


Photo 6

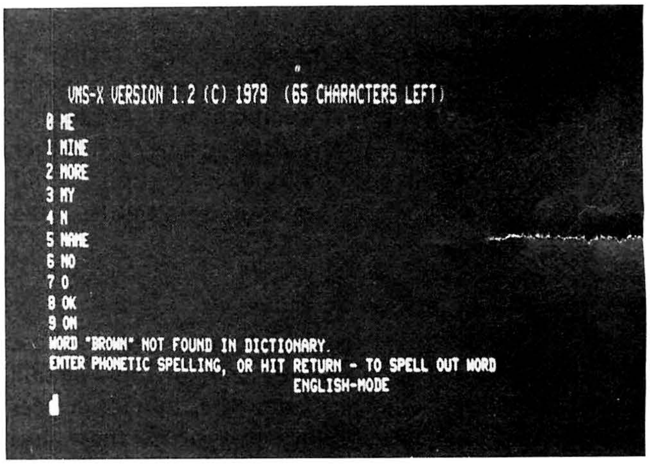


Photo 7

Photo 8

"The bionic man can see fifty miles. With the Bionic Voice, I'll be able to shout fifty miles."

Star combination. Some examples: educational game programs, computer-aided instruction, text editing/-word processing, BASIC language interpreters, and so on.

Another use I have found for the synthesizer is foreign language translation. The English spelling of a word or phrase can be stored in the dictionary, and the foreign language phonetic spelling can be stored with it. When the word or phrase is recalled or typed in, the voice output will be in the foreign language. Unfortunately, it is not capable of a full grammatical and sentence structure translation, but there are some interesting possibilities.

One idea came from a friend who is a registered nurse. She suggested that the translation might be useful in the emergency room of a hospital as a way of asking simple questions like "Where does it hurt?" in other languages. The response to many such questions would have to be gestures or a yes or no nod, but at least the questions could be asked.

Bionics and the Future

Bionics, by definition, is the study of systems whose functions are based upon living systems. It could be said that automobiles and airplanes are bionic devices because they simulate biological systems that have evolved in nature. I have named the voice synthesizer the "Bionic Voice" because that is exactly what it is. We generally think of the capabilities of the human body. The distinction between bionics and tools is vague. Tools are extensions to the human body created through technology. Bionics deals with the simulation of living systems. If we build a bionic arm and attach it to an amputee, do we call the new arm a tool or a bionic replacement?

Similarly, computers have been called extensions of the human mind. In many respects, they simulate living systems, and may one day be classified as a life form, yet we refer to them as tools. Perhaps the ultimate distinction between tools and bionics (when used by humans) is whether or not they extend a capability, or replace something that was lost due to accident or disease.

In this case, Bill has had cerebral palsy since birth. With the aid of high technology, replacements for lost functions can be built. Yet, we need not stop with conventional voice synthesis. We can make it better and more sophisticated.

Bill once said, "The bionic man can see fifty miles. With the Bionic Voice, I'll be able to shout fifty miles."

In a sense, we can all do that right now. Radio, telephones, and television allow our voices to be heard at incredible distances. The difference is that we are using a tool to amplify and transmit our voice. With Bill, the synthesized voice is already in a form suitable for transmission. Our world-wide communications network will become his extended nervous system. Bill will have something that many journalists would love to have: the potential for a direct link into the wire services, the newsroom word processing equipment, and even radio or television.

With the addition of a text editor and text formatter program to the Bionic Voice computer, Bill could conceivably go out in the field, cover a story, edit, and transmit finished copy directly over phone lines, completely bypassing the pad and pencil step. The story is sent directly to the editor's desk without intervening steps, making him the fastest reporter in the world.

From responding with a yes or no nod of the head, Bill went to the slow, clumsy headstick for expression. Each revolution in communications has been a quantum leap forward for Bill. Now he will have the Bionic Voice. He will be able to speak to the world and the world will listen. I cannot begin to imagine what uses Bill will find for his new voice, but if past accomplishments are any indication of things to come, I want to be around in five or ten years to see the results of the seed we have planted. ■

References

Gerardin, Lucien, *Bionics*, trans, Pat Priban, (New York NY, 1968).

Rice, D Lloyd, "Friends, Humans, Countryrobots, Lend Me Your Ears", *BYTE*, Number 12, August 1976.

Tanenbaum, Andrew S, *Structured Computer Organization*, Prentice-Hall, Inc, Englewood Cliffs NJ 1976.

About the Author

Mark Dahmke is employed at the University of Nebraska Computer Network as a programmer/analyst in Academic Computing Services. He has been involved with both large and small computers since 1974. His main interests are writing, systems programming, operating systems design, bionics, voice synthesis, and graphics. He is a senior in Computer Science and owns a personal computer.

BOOK REVIEW

Two Collections of BASIC Programs

Some Common BASIC Programs
by L Poole and M Borchers
Osborne/McGraw-Hill, Inc.
Berkeley CA, 1977
$9.50 192 pp
BASIC Software Library
Volumes 1-5
Scientific Research Inst.
Key Biscayne FL
Reviewed by John A Lehman

There are several available collections of BASIC programs. Many of these are collections of games. The books I am going to discuss are not concerned with games, but rather are collections of general programs. These books are *Some Common BASIC Programs* by Poole and Borchers, and the *BASIC Software Library* by Scientifc Research Inst.

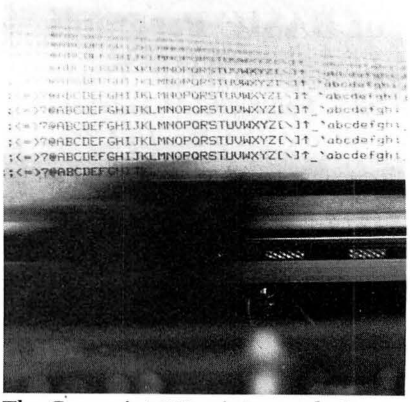
The programs in *Some Common BASIC Programs* deal with mathematics, finance, statistics, and a few miscellaneous topics. There are 75 programs in all, which comes to about 10 cents per program. For each program there is a short description, a sample run, and a listing in Wang BASIC. Wang BASIC is quite close to most microcomputer BASICs; every program in this book will run unchanged in TDL or Microsoft BASIC (which includes the BASIC for the Commodore Pet personal computer). There is not much filler in this book; it is all meat. I have used about 25 of the programs in this book, and all but one of them work very well. The exception is the linear programming program. The fact that it does not work as described is pointed out in the errata sheet for the book. If you want to do mathematics or statistical programming on your computer, this book is a must.

The Scientific Research BASIC Software Library is in several volumes; I can only comment on volumes 1 thru 5. These programs are written in a version of Dartmouth BASIC which is fully described to allow modification of the programs. This is fortunate, since a fair number of these programs will not run on most microcomputers without minor modifications.

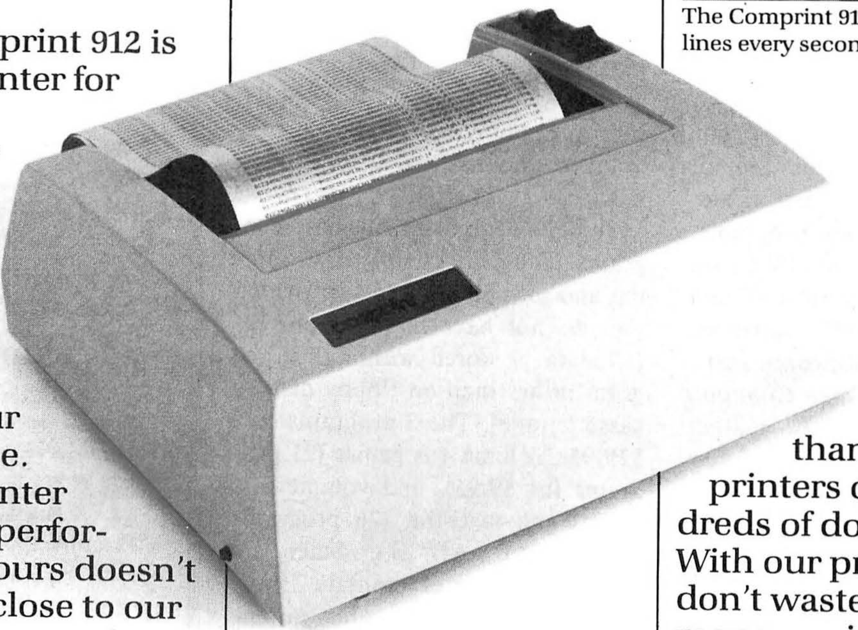
Volume 1 contains programs about financial mathematics, production engineering, games, and pictures. It contains 48 programs and costs $24.95. If you are involved in any sort of industrial activity, the plant layout model on page 71 is probably worth the price of the book. Volume 2 covers mathematics, engineering, and statistics; 46 programs for $24.95. In my opinion, the statistics programs are not as good as the ones in *Common BASIC Programs*. Volume 3 deals with business programs. These are complete accounting and payroll programs, provided you do not have many employees (all data is stored within the program rather than on floppy disk or cassette tape). The 8 programs cost $39.95. Volume 4 is games (21 programs for $9.95), and volume 5 is games and statistics (26 programs for $9.95). All the Scientific Research volumes contain short descriptions, listings, and sample runs.

To compare the two sets of books, *Some Common BASIC Programs* is much cheaper (10 cents a program versus about 50 cents), the programs are of a more consistent quality, and from my point of view there is more useful material. On the other hand, the Scientific Research books provide programs for a much greater variety of tasks and interests than does *Some Common BASIC Programs*, and some of the programs are excellent. If you want mathematics or statistics programs, get *Some Common BASIC Programs*. If you want engineering or simple business programs, get the Scientific Research books. ■

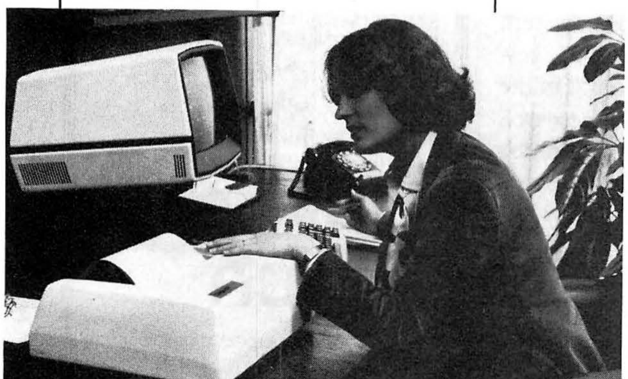
The Comprint 912 printer. No one else can match our speed, our print quality, our quiet operation, or our reliability. Not for $660 they can't.



The Comprint 912 prints nearly 3 lines every second.



Our Comprint 912 is the best printer for the money. Period. Any printer that can match our price can't even begin to match our performance. And any printer that boasts performance like ours doesn't even come close to our price. No matter what your application; computer reports, listings, CRT hard copy, message receiving, scientific/industrial data logging, or anything you can think of, the Comprint 912 is **the** performance leader in printers under $1000.

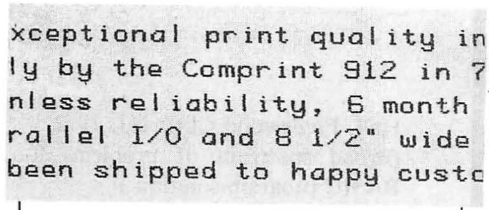


CRT hardcopy is an excellent application for the Comprint 912.

First consider our performance.

Speed. At 225 characters per second (170 LPM) the Comprint 912 is up to 4 times faster than impact printers costing hundreds of dollars more. With our printer you don't waste time and money waiting for your print-out.

Print Quality. Our 9x12 matrix provides sharp, crisp characters. Compare that with our competition. Their very best is a 9x7 matrix, which means no lower case descenders and cramped letters. With the Comprint 912 you don't have to put up with the irritation of fuzzy, hard to read computer printing. This

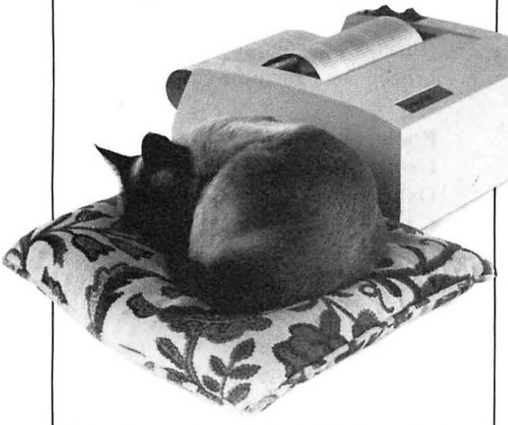


The superior print quality provided by the Comprint 912 is obvious in this actual size sample.

means increased productivity. And because the Comprint 912 makes better originals, our originals make better Xeroxes.

Quiet Operation.

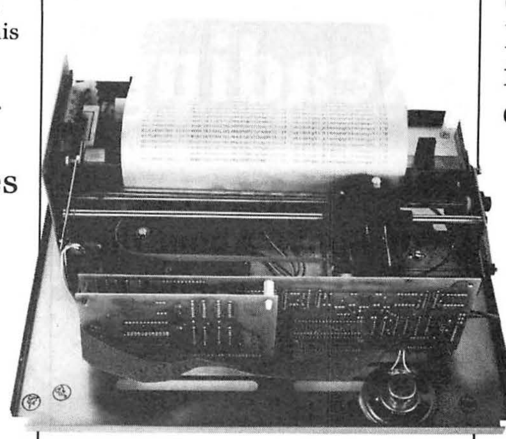
Most computer printers are irritatingly noisy. They can disrupt concentration and reduce the efficiency of anyone working near them. They're noisy because they're


The Comprint 912 is quiet because it's electronic not mechanical.

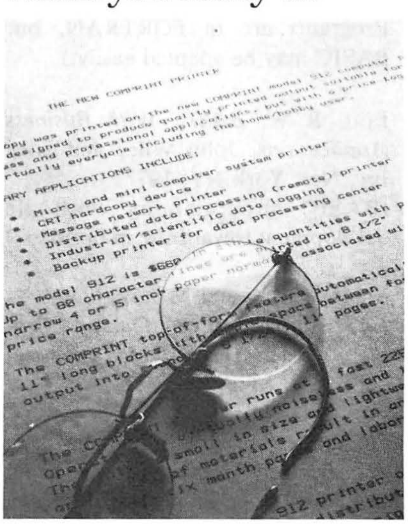
impact. The Comprint 912 has no mechanical print head banging on the paper. It's electronic. It's quiet.

Reliability.

Since the Comprint 912 prints electronically, rather than mechanically like ordinary impact printers, we have fewer moving parts and less vibration. The Comprint 912 has fewer things to go wrong and less wear. That's why we offer a 6 month warranty, twice the industry standard. The key to all this superior performance is our special


Fewer moving parts in the Comprint 912 mean greater reliability.

paper. This aluminized "silver paper" works just like ordinary paper. It won't fade or discolor and actually costs less than plain paper and one time ribbons. For the vast majority of printing applications it's just plain better than plain paper. Especially when you consider the hidden costs of plain paper printers due to their inferior performance compared to the Comprint 912. And on those rare occasions when you really do



need a plain bond paper copy, just run your Comprint 912 printout through your plain bond copy machine and you've got it. Even though our paper is special, it's available everywhere; from your dealer or distributor, or from us.

Now consider our price.

The Comprint 912.

$660 with parallel interface, $699 with serial interface.

We could talk about our other advantages, like our 80-character lines on 8-1/2" wide paper, or our compact, light-weight size, and the fact that the Comprint 912 has no ribbons to mess with, no chemicals, nothing to add but paper.

But you have to see for yourself. Before you buy any printer, insist on seeing the Comprint 912, **the** performance leader, at your local computer store or industrial distributor. Or contact us for a descriptive brochure, a sample print-out, and applications literature.

comprint

The performance leader.

Computer Printers International, Inc.
340 E. Middlefield Rd.
Mountain View, California 94043
415 969-6161

Circle 35 on inquiry card.

Some "BASIC" Reading:

A Short List of Books About the BASIC Language

W B Agocs
Kutztown State College
Kutztown PA 19530

Bennett, W R Jr, *Scientific and Engineering Problem Solving With the Computer*, Prentice-Hall, Englewood Cliffs NJ, 1976.
(Quite comprehensive, with a unique development of the BASIC material. Based on the HP 2116B computer. Extensive use of language analysis. References at end of sections are advanced.)

Bennett, W R Jr, *Introduction to Computer Applications for Non-Science Students (BASIC)*, Prentice-Hall, Englewood Cliffs NJ, 1976.
(An abridgement of above.)

Chirlian, P M, *Beginning BASIC*, Dilithium Press, Portland OR, 1978.
(BASIC procedures using flowcharts and elementary programs.)

* Coan, J S, *Basic BASIC*, Hayden Book Co Inc, Rochelle Park NJ, 1970.
(An interesting development of BASIC for computer programming.)

Coan, J S, *Advanced BASIC*, Hayden Book Co Inc, Rochelle Park NJ, 1977.
(A continuation of *Basic BASIC* — an extension of BASIC with programs.)

Deitel, H M, *Introduction to Computer Programming With the BASIC Language*, Prentice-Hall, Englewood Cliff NJ, 1977.
(BASIC programming developed using flowcharts and programs.)

Dwyer, T A and M Critchfield, *BASIC and the Personal Computer*, Addison-Wesley, Reading MA, 1978.
(An introduction to BASIC using an 8-hour, self-study approach. Excellent.)

Gottfried, B S, *Programming With BASIC*, Schaum's Outline Series, McGraw-Hill Book Co, New York NY, 1975.
(Broad treatment and presentation of BASIC with problem development. Serves a dual purpose of teaching programming as well as some interesting mathematics.)

* Gruenberger, F, *Computing With the BASIC Language*, Canfield Press, San Francisco CA, 1972.
(Basic programming with BASIC. Some interesting problems.)

Hubin, W, *BASIC Programming for Scientists and Engineers*, Prentice-Hall, Englewood Cliffs NJ, 1978.
(BASIC programming with a slight slant toward scientific and engineering utility.)

Kemeny, J G and T E Kurtz, *BASIC Programming*, John Wiley and Sons Inc, New York NY, 1971 (2nd ed).
(The use of BASIC in programming by the developers of the language.)

LaFara, R L, *Computer Methods for Science and Engineering*, Hayden Book Co Inc, Rochelle Park NJ, 1973.
(An excellent text on numerical analysis combined with flowcharts. Programs are in FORTRAN, but BASIC may be adapted easily.)

Lott, R W, *BASIC With Business Applications*, John Wiley and Sons Inc, New York NY, 1977.
(BASIC programming with its development toward use in business.)

Maurer, H A and M R Williams, *A Collection of Programming Problems and Techniques*, Prentice-Hall, Englewood Cliffs NJ, 1972.
(Broad spectrum of problems for BASIC program solution.)

Merrill, J R, *Using Computers in Physics*, Houghton Mifflin Co, Boston MA, 1976.
(Excellent treatment of problems in physics using BASIC and FORTRAN.)

Murrill, P W and C W Smith, *BASIC Programming*, Intext Educational Publishers, Scranton PA, 1971.
(Programming in BASIC.)

Peckham, H D, *Computers, BASIC, and Physics*, Addison-Wesley, Reading MA, 1971.
(Programming with BASIC of the mathematics of physics and of physical applications.)

Pegels, C C and R C Verkler, *BASIC With Business and Management Applications*, Holden-Day Inc, San Francisco CA, 1976.
(Business programs using BASIC procedures. Well-developed.)

Poole, L and M Borchers, *Some Common BASIC Programs*, Osborne/McGraw-Hill, Berkeley CA, 1977.
(Programs in BASIC oriented toward business, such as taxes, interest, etc.)

Rice, J R, *Introduction to Computing With BASIC*, Holt, Rinehart, and Winston Inc, New York NY, 1973.
(A text development of BASIC programming.)

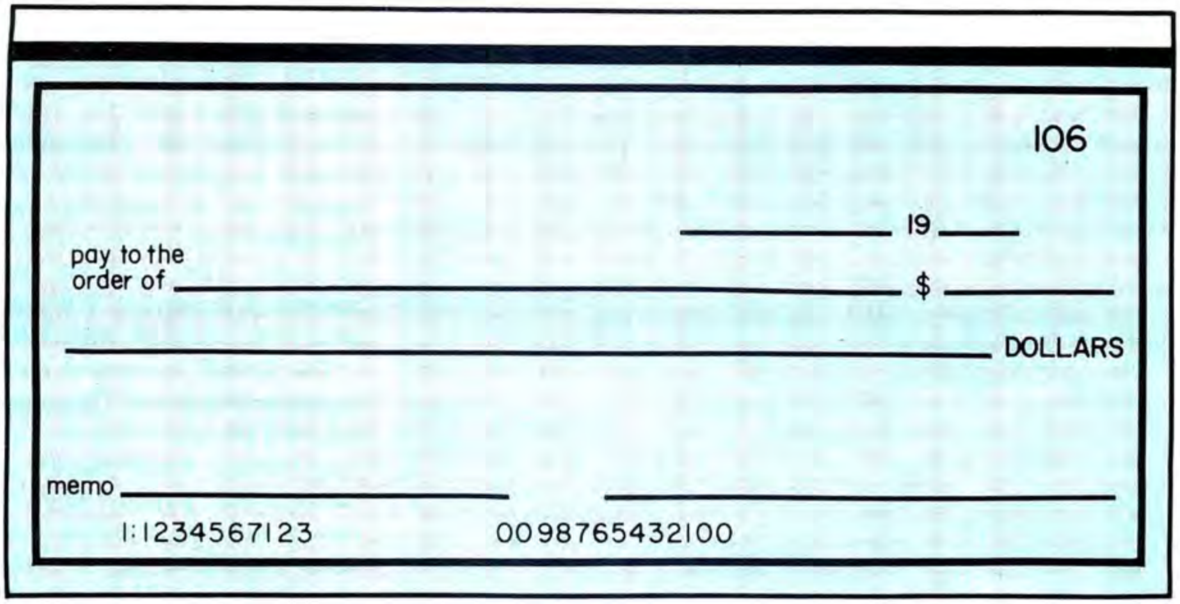
Singer, B M, *Programming in BASIC, With Applications*, McGraw-Hill Book Co, New York NY, 1973.
(BASIC programming. Well-developed, but limited.)

* Smith, R D, *Discovering BASIC*, Hayden Book Co Inc, Rochelle Park NJ, 1970.
(Elementary programming with programs in BASIC.)

Worth, T, *BASIC for Everyone*, Prentice-Hall, Englewood Cliffs NJ, 1976.
(A comprehensive treatment of BASIC programming.)

[References marked with an asterisk (*) are particularly suitable for the beginning BASIC programmer.—ed]

The Personal Check Writer



by Andrew A Recupero

The personal check writer program presented here will not only put your computer to work, but it will also take some of the drudgery out of paying monthly bills.

Basically, the program steps through a list of your creditors. The programmer enters the amount wanted on the check. After having checked all entries, the program fills out each check, one by one. All the user has to do is sign them. When finished, the computer gives the programmer the final checkbook balance.

I decided to write this program while paying my monthly bills. Paying bills is not exciting to begin with; doing basically the same thing every month began to annoy me. To top it off, I always seemed to get hung up on spelling forty and fourteen. I had just bought a personal computer, and knew that this was a job it could handle with ease. Unfortunately, I had to wait for a hard copy output device which is a must for this application. After getting a used Selectric typewriter equipped to print under command of the computer, I wrote the program, and have been using it to pay my monthly bills ever since.

Why not use your personal computer to print out your checks every month?

The program is written in BASIC and, in my case, runs on an 8080 system with a keyboard and the Selectric output device. The program occupies about 12 K bytes of memory (this includes the computer's BASIC interpreter) and should be adaptable to any system with a hard copy device and a BASIC package which can manipulate string variables. [A string is a collection of letters, numbers, and/or symbols which can be manipulated by the computer. Consult the software manual of your computer, or your local computer store to see whether or not your BASIC package can handle character strings...ed]

Using the Program

After loading the program and entering RUN, the programmer is prompted for the check date, year, beginning checkbook balance, and beginning check number. If you have prenumbered checks, enter a 0 for the check number.

The program then enters the input amount phase. In this phase the program steps through a table containing the names of creditors. After each name is displayed, the program waits for the user to input the amount that is wanted on the check. The decimal point can be omitted to allow faster data entry and to reduce errors. Thus an amount of $31.59 could be entered as 3159. The text equivalent of the dollar amount is displayed after each input. This

Listing 1: Sample run of the program in listing 2 except for the actual printing of the checks.

```
RUN
* * * P E R S O N A L   C H E C K   W R I T E R * * *

MONTH, DAY OF CHECK (EX. SEPT 6)? MAY 10
YEAR (EX. 1978 OR 78)? 79
STARTING CHECK NUMBER (0 IF PRENUMBERED)? 213
BEGINNING CHECK BOOK BALANCE? 287.55

   INPUT OPTIONS
   1 .   CHECK AMOUNT (INCL CENTS, DECIMAL OPTIONAL)
   2 .   OMIT - TO OMIT NAME
   3 .   OK - TO LEAVE AS IS
   4 .   REDO - TO RESTART INPUT AMOUNT PHASE
   5 .   LIST - TO LIST PAY TO AND AMOUNT
   6 .   DONE - TO START OUTPUT PHASE

BANK OF AMERICA 0 AMOUNT = ? 9313
NINETY-THREE AND 13/100

BROWNING-FERRIS INDUSTRIES 0 AMOUNT = ? OMIT

CHEVRON 0 AMOUNT = ? 4700
FORTY-SEVEN AND NO/100

THE EMPORIUM 0 AMOUNT = ? DONE

   BANK OF AMERICA 93.13
   BROWNING-FERRIS INDUSTRIES -OMIT-
   CHEVRON 47.00
   THE EMPORIUM -OMIT-
   D & J PEST CONTROL -OMIT-
   MACY'S -OMIT-
   MERVYN'S -OMIT-
   NORTHERN CALIFORNIA SAVINGS -OMIT-
   PACIFIC, GAS & ELECTRIC -OMIT-
   PACIFIC TELEPHONE -OMIT-
   SAN JOSE WATER WORKS -OMIT-
   SEARS, ROEBUCK AND CO -OMIT-
ENTER 'OK' OR 'REDO' TO PROCEED? OK

P R I N T   C H E C K   P H A S E
   SPECIAL SETUP INSTRUCTIONS:
   LEFT MARGIN AT 0
   TOP OF CHECK TO EDGE OF PLASTIC GUIDE
   SINGLE SPACING

PRESS ANY KEY WHEN EACH CHECK IS READY
ENDING BALANCE IS $147.42
   E N D   O F   J O B
OK
```

gives the programmer a chance to double-check the input. If you do not want a particular check printed, enter OMIT.

At this point, four additional input commands are accepted in place of the dollar amount. These are REDO, OK, LIST, and DONE. The REDO command restarts the input amount phase from the beginning. This permits the programmer to change a previous entry if it is wrong, or if there are second thoughts about how much you want to remit. The previously entered amount is displayed; if it does not need to be changed, enter OK.

The DONE command ends the input amount phase. This command is used if the user wants to skip stepping through the remainder of the names in the "pay-to" table. All the names have their check amount value initialized to 0, which is treated as "omit." Therefore, any name not given a check amount will be omitted from printing as a default. The input amount phase automatically ends after the last name in the pay-to list is processed.

The LIST command lists all the names in the pay-to table and their check amounts. The user can use this command to check any previous entry for errors.

When the input amount phase ends, the program performs an automatic LIST command and then waits for either an OK or REDO input. The OK input starts the print-check phase.

The print-check phase first lists some instructions on how to set up the printer. The user will have to customize this for the printer. I found this to be a very useful way for remembering the setup procedure from one session to the next. Using the names in the pay-to table that were not to be omitted, the program waits for a go signal, processes one check, ejects the check, and then waits for the next go signal. This allows enough time to get the next check set up. After the last check has been processed, the next go signal causes the ending balance to be output.

A sample run of the program up to the print check phase is shown in listing 1.

Installing the Program

Listing 2 is the BASIC program. Before the programmer can run the program, there are some changes that will have to be made (aren't there always!). The majority of these changes are to customize the program to your checks.

The first thing to change is the pay-to names in lines 300 to 340. The names shown are the ones that I use. Statements may be added or omitted if necessary. Next, count the number of individual names you have, and set the variable DNUM equal to this value (line 190). This will cause the DIM statements and FOR loops to initialize to the proper number of entries. The user might notice that the zero element of the arrays is not used. I chose this approach to simplify using this program with BASIC interpreters which do not allow a zero index.

One comment about the listing is in order. Note that I have used several three- and four-letter variables. The BASIC package I use allows this as long as the first two characters are unique. I find the longer variable names make the program easier to read, but you can shorten them to two characters if your BASIC has different restrictions.

The next step requires a little more work. This involves determining the starting column number for each item to be printed on the check. There are six such items: the date, year, check number, pay-to name, amount, and the text form of the amount. There are two ways of going about this. The first is to use my values and then adjust them by trial and error. The other is to print out a "ruler" which can be laid over the check, and will read off the proper columns. A sample BASIC program to create the ruler is given in listing 3. The important thing is to pick a left-edge reference on the printer that will be the same for the ruler and the check. Write this

Listing 2: List of the Personal Check Writer program.

```
LIST.
10 REM * * * P E R S O N A L  C H E C K  W R I T E R * * *
20 REM REVISION 7
30 REM
40 REM SET STRING SPACE
50 CLEAR 100
60 REM DEFINE COLUMN POSITIONS FOR:
70 REM      CHECK NUMBER
80 NUC=52
90 REM      DATE
100  DATC=36
110  REM      YEAR
120  YC=45
130  REM      PAY TO
140  PC=9
150  REM      AMOUNT TEXT
160  AC=4
170  REM      DOLLAR AMOUNT
180  DOLC=49
190  REM SET DNUM = NUMBER OF PAY TO ENTRIES
200  DNUM=12
210  DIM PAY2$(DNUM),AMT$(DNUM)
220  DIM UN$(9),TEN$(8),EEN$(10)
230  REM INITIALIZE PAY TO TABLE
240  FOR M=1 TO DNUM
250  READ PAY2$(M)
260  REM SET AMT TO DEFAULT TO OMIT
270  AMT$(M)="0"
280  NEXT M
290  REM  -- PAY TO TABLE --
300  DATA "BANK OF AMERICA","BROWNING-FERRIS INDUSTRIES"
310  DATA "CHEVRON","THE EMPORIUM","D & J PEST CONTROL"
320  DATA "MACY'S","MERVYN'S","NORTHERN CALIFORNIA SAVINGS"
330  DATA "PACIFIC, GAS & ELECTRIC","PACIFIC TELEPHONE"
340  DATA "SAN JOSE WATER WORKS","SEARS, ROEBUCK AND CO"
400  REM INITIALIZE UNITS, TENS AND TEENS ARRAYS
410  FOR M=1 TO 9: READ UN$(M): NEXT M
420  DATA ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE
430  FOR M=1 TO 8: READ TEN$(M): NEXT M
440  DATA TWENTY,THIRTY,FORTY,FIFTY,SIXTY,SEVENTY,EIGHTY,NINETY
450  FOR M=1 TO 10: READ EEN$(M): NEXT M
460  DATA TEN,ELEVEN,TWELVE,THIRTEEN,FOURTEEN,FIFTEEN,SIXTEEN
470  DATA SEVENTEEN,EIGHTEEN,NINETEEN
480  REM
490  PRINT"* * * P E R S O N A L  C H E C K  W R I T E R * * *"
500  PRINT
510  INPUT "MONTH, DAY OF CHECK (EX. SEPT 6)";DATE$
520  INPUT "YEAR (EX. 1978 OR 78)";YR$
530  IF LEN(YR$) > 2 THEN YR$=RIGHT$(YR$,2)
540  INPUT "STARTING CHECK NUMBER (0 IF PRENUMBERED)";CKN
550  INPUT "BEGINNING CHECK BOOK BALANCE"; BL
560  PRINT
570  REM START OF INPUT AMOUNT PHASE
580  REM PRINT INPUT INSTRUCTIONS
590  GOSUB 3100
600  FOR M=1 TO DNUM
610  PRINT
620  PRINT PAY2$(M) SPC(2) AMT$(M) SPC(2);
630  INPUT "AMOUNT = "; DOL$
640  REM
```

Listing 2 continued on next page

Listing 2 continued:

```
650  IF DOL$= "OMIT" THEN AMT$(M)="0": GOTO 950
660  IF DOL$= "OK" THEN 950
670  IF DOL$= "DONE" THEN PRINT: GOTO 970
680  IF DOL$= "REDO" THEN 600
690  IF DOL$= "LIST" THEN GOSUB 4100: GOTO 620
700  REM CHECK INPUT FOR DECIMAL AND REMOVE IT
710  L=LEN(DOL$)
720  FOR J=1 TO L
730  IF MID$(DOL$,J,1) = "." THEN 770
740  NEXT J
750  GOTO 810
760  REM MUST HAVE 2 DIGITS AFTER DECIMAL
770  IF L-J<> 2 THEN 870
780  IF J=1 THEN DOL$=RIGHT$(DOL$,2): GOTO 810
790  DOL$=LEFT$(DOL$,J-1) + RIGHT$(DOL$,2)
800  REM CHECK FOR VALID INPUT
810  IF LEN(DOL$)>6 THEN PRINT SPC(2)"* AMOUNT
     TOO LARGE *":GOTO 620
820  FOR J=1 TO LEN(DOL$)
830  K$=MID$(DOL$,J,1)
840  IF K$<"0" OR K$>"9" THEN 870
850  NEXT J
860  GOTO 910
870  PRINT SPC(2) "* * INPUT ERROR * *"
880  GOSUB 3100: REM PRINT INSTRUCTIONS
890  GOTO 620
900  REM ERROR INPUT IS SINGLE DIGIT
910  IF LEN(DOL$)=1 THEN 870
920  AMT$(M)=DOL$
930  REM OUTPUT TEXT OF AMOUNT AS VISUAL CHECK
940  GOSUB 5100
950  NEXT M
960  REM          DONE WITH INPUT — DO LIST
970  GOSUB 4100
980  INPUT "ENTER 'OK' OR 'REDO' TO PROCEED"; DOL$
990  IF DOL$ = "REDO" THEN 600
1000 IF DOL$ <> "OK" THEN 980
1010 PRINT
1020 PRINT "P R I N T  C H E C K  P H A S E"
1030 REM
1040 PRINT TAB(4) "SPECIAL SETUP INSTRUCTIONS:"
1050 PRINT TAB(6) "LEFT MARGIN AT 0"
1060 PRINT TAB(6) "TOP OF CHECK TO EDGE OF PLASTIC GUIDE"
1070 PRINT TAB(6) "SINGLE SPACING"
1100 PRINT
1110 PRINT "PRESS ANY KEY WHEN EACH CHECK IS READY"
```

Listing 3: List and run of a sample program to generate a "ruler" for finding the appropriate columns for entries on your check.

```
1120 REM
1130 REM          START OF OUTPUT CODE
1140 REM
1150 FOR M=1 TO DNUM
1160 REM SKIP IF AMOUNT IS ZERO
1170 IF VAL(AMT$(M))=0 THEN 1470
1180 REM WAIT FOR "GO" SIGNAL (ANY KEY)
1190 WAIT 6,2,2
1200 REM INDEX TO CHECK NUMBER LINE
1210 IF CKN>0 THEN PRINT TAB(NUC) CKN: CKN=CKN+1
```

reference into the program as special setup instructions, lines 1040 thru 1070. The column numbers are treated as variables in the program, and are initialized in statements 80 thru 180. Put the values in these statements.

Next, the user must set up for the number of blank lines between the different check entries. For my checks, all of the entries are contained on four lines. The check number is on one, the date and year are on another, the pay-to name and amount are on the third line, and the text of the amount is on the fourth line.

The easiest way to determine the various spacings is to put a check into the printer, establish a convenient top of check reference, and manually rotate the platen, noting the number of lines between the various entries. Remember that each existing PRINT statement produces one line feed. You will need to add or delete PRINT statements only if extra or fewer line feeds are required. The check number is printed by the statement in line 1210. Note that the second statement on line 1210 increments the check number. In my computer's BASIC package, this statement will be executed only if the IF statement is true. Not all BASICs operate this way, so be careful. Line 1220 simulates printing the check number for the prenumbered option. If all of the checks are prenumbered, you can omit lines 80, 540, 1210 and 1220. The date and year are printed by line 1250. The pay-to name and amount are printed by lines 1290 and 1310 thru 1340, and the text of the amount is printed by the subroutine call at line 1380.

Lines 1400 thru 1430 cause a number of line feeds to eject the check. This may have to be changed to suit check size.

The last change involves the method of signaling "go" to the computer between each check. I used the WAIT statement, which is set up to wait for a data ready input from my keyboard. The problem I am trying to solve here is to tell the

```
1220  IF CKN=0 THEN PRINT
1230  REM INDEX TO DATE LINE (IF NECESSARY)
1240  REM
1250  PRINT TAB(DATC) DATE$ TAB(YC) YR$
1260  REM INDEX TO PAY TO LINE
1270  PRINT
1280  PRINT
1290  PRINT TAB(PC) PAY2$(M);
1300  REM SKIP DOLLARS PART IF < 99 CENTS
1310  IF LEN(AMT$(M))=2 THEN PRINT
      TAB(DOLC)"*";:GOTO 1350
1320  PRINT TAB(DOLC)"*"
      LEFT$(AMT$(M),LEN(AMT$(M))-2);
1330  REM PRINT CENTS PART
1340  PRINT "." RIGHT$(AMT$(M),2)
1350  REM INDEX TO AMOUNT TEXT LINE
1360  PRINT
1370  PRINT TAB(AC);
1380  GOSUB 5100: REM PRINT TEXT FORM OF
      AMOUNT
1390  REM    EJECT CHECK
1400  PRINT
1410  PRINT
1420  PRINT
1430  PRINT
1450  REM CALCULATE NEW BALANCE
1460  BL=BL - VAL(AMT$(M))/100
1470  NEXT M
1480  REM ALL CHECKS WRITTEN
1490  REM WAIT FOR "GO" AND PRINT ENDING
      BALANCE
1500  WAIT 6,2,2
1510  PRINT "ENDING BALANCE IS $";BL
1520  PRINT TAB(10) "E N D  O F  J O B"
1530  GOTO 9999
3000  REM
3010  REM PRINT INPUT OPTIONS SUBROUTINE
3020  REM
3100  PRINT TAB(2) "INPUT OPTIONS"
3110  PRINT TAB(4) "1. CHECK AMOUNT (INCL
      CENTS, DECIMAL OPTIONAL)"
3120  PRINT TAB(4) "2. OMIT — TO OMIT NAME"
3130  PRINT TAB(4) "3. OK — TO LEAVE AS IS"
3140  PRINT TAB(4) "4. REDO — TO RESTART IN-
      PUT AMOUNT PHASE"
3150  PRINT TAB(4) "5. LIST — TO LIST PAY TO
      AND AMOUNT"
3160  PRINT TAB(4) "6. DONE — TO START OUT-
      PUT PHASE"
3170  PRINT
3180  RETURN
4000  REM
4010  REM LIST PAY TO AND AMOUNT
      SUBROUTINE
4020  REM
4100  PRINT
4110  FOR J=1 TO DNUM
4120  PRINT SPC(5) PAY 2$(J) SPC(2);
4130  IF VAL(AMT$(J))=0 THEN PRINT
      "—OMIT—": GOTO 4170
4140  IF LEN(AMT$(J))=2 THEN 4160
4150  PRINT LEFT$(AMT$(J),LEN(AMT$(J))-2);
4160  PRINT "." RIGHT$(AMT$(J),2)
```

Listing 3 continued on next page

Listing 3 continued:

```
4170    NEXT J
4180    RETURN
5000    REM
5010    REM CONVERT AMT(M) TO TEXT
        SUBROUTINE
5020    REM
5100    AMT = VAL(AMT$(M))/100
5110    REM CHECK IF CENTS ONLY
5120    IF AMT < 1 THEN 5390
5130    N = INT(AMT/1000)
5140    IF N > 0 THEN PRINT UN$(N) " THOUSAND ";
5150    AMT = AMT - 1000*N
5160    N = INT(AMT/100)
5170    IF N > 0 THEN PRINT UN$(N) " HUNDRED ";
5180    AMT = AMT - 100*N
5190    N = INT(AMT/10)
5200    IF N > 1 THEN 5290
5210    REM AMOUNT LEFT IS LESS THAN $20
5220    IF N < 1 THEN 5320
5230    REM BETWEEN $10 - $19 LEFT
5240    AMT = AMT - 10
5250    N = INT(AMT)
5260    PRINT EEN$(N + 1) " ";
5270    GOTO 5350
5280    REM $20 - $99 LEFT
5290    PRINT TEN$(N - 1);
5300    IF INT(AMT - 10*N) > 0 THEN PRINT " - ";:GOTO 5320
5310    PRINT " ";
5320    AMT = AMT - 10*N
5330    N = INT(AMT)
5340    IF N > 0 THEN PRINT UN$(N) " ";
5350    AMT = AMT - N
5360    PRINT "AND ";
5370    IF AMT = 0 THEN PRINT "NO/100" : RETURN
5380    REM PRINT CENTS
5390    PRINT RIGHT$(AMT$(M),2) "/100"
5400    RETURN
5410    REM
9999    REM END OF PROGRAM
OK

LIST.

20 REM SET A$ TO NUMBERS FOR RULER
30 A$ = "1234567890"
40 FOR M = 1 TO 6
50 PRINT SPC(9) MID$(A$,M,1);
60 NEXT M
70 PRINT
80 FOR M = 1 TO 6
90 PRINT A$;
100  NEXT M
110  PRINT
120  PRINT "< - LEFT EDGE HERE
OK
RUN

          1         2         3         4         5         6
123456789012345678901234567890123456789012345678901234567890
< - LEFT EDGE HERE

OK
```

program that I am ready without having the typed signal printed on my check. In my system, before I signal the first go, I use a front panel switch to direct all display output to the printer. Your system may not have this problem. Each BASIC usually has some way of dealing with this type of problem. If necessary, and if you have the expertise, you could write a user function in machine code which would wait for the go signal and which would be interfaced to the program with a USR statement. The WAIT statements are in lines 1190 and 1500.

If you are lucky enough to have a printer with tractor feed, and have personal checks on continuous forms, then you can change the program to print all the checks without waiting for the intervening go signals.

When you have made all the changes, I suggest a trial run on a blank piece of paper cut to the size of the checks. Hold this up to a light with one of the checks behind it, and visually check that all is as planned. If everything is fine, you are ready to go.

Conclusions

After using this program for awhile, I have discovered some enhancements that you might like to add. One allows the addition of an optional memo to the check. Another allows a few new names to be added to the pay-to table while running the program, to accommodate those once-in-a-while checks. Yet another prints out a summary which could be taped into the check register. By far the next most useful project would be to devise a way to have the computer sign your name, stuff the envelope, and add the stamp. I'll leave that for those who are interested in robotics.

Paying bills is still not one of my big joys, but using this program to write out my checks leaves me a lot less frazzled, and also provides me with a good response when someone asks, "What do you do with your computer?" ■

Product Evaluation: Heathkit H8

by Mark Snow

A few months ago I was given an unusual opportunity for a sixteen-year-old: onComputing sent me a Heathkit H8 digital computer kit to assemble, operate, and analyze. This article is a report on my progress with the computer, as well as a discussion of some of the problems I encountered while constructing it.

The H8 contains hundreds of electronic parts mounted on printed circuit boards. All of the main boards plug into the *mother board,* so called because it literally holds the "family" of boards together. The mother board was the first part of the computer that I put together.

The most difficult part of building the kit was learning how to solder well, although I learned quickly after soldering the mother board's 500 connections.

The next section to be assembled was the mainframe (the main portion of the computer) and part of the power supply. Heath has put much thought into the design of their power supply, which contains excellent filtering to reduce electronic noise. The printed circuit cards are made from epoxy glass; the chassis has a pair of rugged black plastic side panels held together by a sheet metal back, base, and front support. The power supply transformer fits neatly in the rear of the chassis. A large filtering capacitor and a switch box for the power supply are located in the rear of the unit. The bottom is a perforated panel that allows air circulation throughout the unit. The front grill of the chassis is a light gray, preformed piece of sheet metal with punchouts for a key pad and light-emitting diode (LED) display. There is also a hefty sheet metal cover with raised vents for cooling.

The last part of the H8 to be constructed is the control circuit board. This board contains a key pad for control of the H8, as well as a nine-digit LED display for monitoring register and memory content. This control circuit board is by far the most complicated section of the assembly. Several integrated circuits and transistors have to be soldered to the double-sided printed circuit board. Key pad switches and LEDs also have to be soldered to this board. Heath supplies all of the necessary solder for the kit.

The most difficult part of building this board involves the wiring harness, a group of wires prepared in advance and tied together in a bundle for quick installation. In this method, all wiring from the mother board to the control board is contained in two 25-pin connectors. I had some trouble soldering individual wires to clips in the harness sockets. Although it was a little clumsy for me, this harness method is still the easiest method for wiring a kit such as this. The assembled control board is attached to the H8 chassis in the front of the unit. All of the basic H8 assembly is complete at this point, except for the installation of the central processing unit.

The central processing unit is a preassembled circuit board that plugs directly into the H8's mother board. It is factory tested for reliability. An 8080-A microprocessor is used. Also on this board is a programmable memory circuit. The company has supplied a schematic and parts list of the central processing unit. If any problems occur in this board within the warranty period (ninety days), free repairs will be made. If out-of-warranty boards fail, the user has enough information on the board to attempt to repair it.

With the completion of the mother board, control board, power supply, mainframe, and the insertion of the central processing unit, the H8 computer is completed. To operate the computer, an 8 K byte memory board (H8-1) has to be added with a minimum of 4 K bytes of memory. The assembly of the memory board is straightforward. Eighteen-pin sockets are soldered in place to hold the circuits, and ten more sockets are put in place for various other integrated circuits. The remaining section of this board is the voltage regulation circuitry, consisting of two regulators and several small capacitors together with one or two resistors. Little work is needed to assemble this board.

With just one H8-1 memory board completed, the H8 computer can be up and running. Heath in-

cludes steps to test the whole computer at every step of the way during the construction process. Starting with the mother board and power supply, a voltmeter is used to test the circuitry. The control board and central processing unit are tested with built-in LEDs that light to show that the respective boards are functioning. With the memory boards added and some entry of data, the front panel display reads "Your H8 is up and running."

When the H8 was finally in operation, I decided to complete a second H8-1 memory board followed by an H8-5 serial and cassette board. H8-5 was a lot of fun to assemble because it contained so many different types of components, such as voltage regulators, resistors, capacitors, transistors, optic isolators, diodes, a variety of integrated circuits, and two relays. Construction went smoothly on the serial board. Many jumper wires are used to add versatility in connecting the H8 with many types of serial devices. The data transfer rate is adjustable from 110 to 9600 bits per second (bps).

The cassette portion of this board is also versatile. The data transfer rate is adjustable. The output level to the audio recorder can be altered to accommodate MIC connections, or AUX connections. The adjustments and testing of this board were easy because of an LED testing circuit. The builder adjusts two potentiometers until an LED either comes on or blinks off. One nice feature of the H8-5 board is its ability to use either one or two cassette recorders. When two are used, one recorder works for loading and the other for dumping information. For text editing or working on long programs, two recorders cut down on the rewinding and erasing of tape.

The next I/O (input/output) board constructed was an H8-2 parallel board. This board contains three independent, parallel ports with complete handshaking capabilities. The port addresses of this device are user-selectable for specialized application. Extremely good documentation is supplied with this board. Information on programming the channels and how they work is supplied in the construction manual. The parallel board was not too difficult to assemble, although I had some problems with solder bridges and socket pins breaking. These problems were generated by working at 2:00 AM when I was half asleep. Working when you're tired doesn't pay; take your time and work on a kit only when you're alert. The parallel board is an ideal device for experimenters who wish to control motors or relays with their computers, or run devices such as printers or paper-tape readers.

In addition to the H8 computer, I was supplied with a WH 17 floppy disk system. The entire disk operating system, control board, single disk drive unit, and system software came as a complete package, with all hardware fully assembled. It was only a five-minute operation to install the control board in the H8 mainframe and connect one ribbon cable between the disk housing and the control board. But in those five minutes the amount of memory space was multiplied by about ten. A single floppy disk drive added 102 K bytes of memory to the computer. The drive itself is a Wanco-82, although the control board is manufactured by Heath. The disk system utilizes a hard sec-

Kitmanship

by onComputing Staff

The story of Mark Snow's assembly and usage of the Heathkit H8 computer demonstrates some of the advantages of entering personal computing via the kit route as mentioned by Elizabeth Hughes in her article in the Summer 1979 issue of onComputing.

onComputing visited Mark recently. The neat construction area at his house was filled with back issues of electronics and computer magazines sitting on the shelf next to a well-used soldering iron. What prompts a sixteen-year-old to build a computer kit? In Mark's case it runs in the family. His grandfather was a lifelong tinkerer and inventor, and Mark started building things with his Erector set when he was five years old.

Over the years Mark has progressed to various types of electronics kits, often ending up by modifying the project to suit his own needs. With this background, it's not surprising that he plunged into the computer kit project without a qualm.

Despite his previous experience in electrical kit construction, Mark had had little experience with computers. The ease with which the project was completed, and the benefits he derived from building his own computer, should have a heartening effect on anyone contemplating entry into personal computing by kit assembly.

Reducing Mark's project to basic facts and figures, he assembled the Heathkit H8 computer (two H8-1 memory boards, an H8-2 parallel interface board, and an H8-5 serial input/output and cassette interface) in less than one hundred working hours spread over five or six days. He completed his system with a ready-to-use dual floppy disk drive and an H9 video terminal.

Mark gives high grades to the Heathkit manuals which accompany the kit, and also to the responsiveness of the Heath employees in solving minor problems that arose. For example, an initial inventory of the kit revealed that a few minor parts were missing and one part was damaged. A phone call to the Heath customer

tor 5¼ inch diskette as its storage medium. This is a common diskette available at many stores.

A few weeks after I completed the H8 system, I received an assembled H9 video display terminal. This terminal is user-adjustable. A serial data transfer rate is available from a range of 110 to 9600 bps. There are also special features that add to the terminal's versatility, including a plot mode to allow graphs to be generated. Simple figures can also be produced. Two drawbacks are a lack of graphics, and the printing of only uppercase letters. Nevertheless, this is a fine device; it's quite a buy at $550. The H9 can display twelve lines of eighty characters. There is an automatic scrolling feature and complete cursor control. After reading the H9 construction manual, the assembly of this kit would be fairly simple but time-consuming.

The Heath H8 system is designed around the BASIC language. Heath has two versions of BASIC — Benton Harbor and Extended. The Benton Harbor version requires 8 K bytes of memory; the Extended version requires 12 K bytes, although 16 K bytes are preferred. A feature of interest to nontypists is the built-in function that allows you to type only the first two or three characters of a command and have the computer finish the statement. For example, if you type "PRI," the machine will respond with "PRINT."

Heath supplies an assembler and a text editor for programming. [See "Understanding Software" in this issue.] I have only begun to understand how to program machine code and am therefore unable to do justice to this software by commenting on it. One other software package is a debugger, which is an enhanced version of the system front panel. This program is used to debug user machine code.

After I became familiar with the hardware and software in the H8, I began experimenting with motor controls and robotic simulations. I next acquired a Terrapin Turtle Robot to experiment with in motor controls. I wrote a simulation which takes a maze of data points on a matrix and simulates obstacles in the path of the turtle. As the turtle moves around on a tabletop, data points acting as obstacles force the robot to turn and change directions. As this process is taking place, a record of the turtle's movement is stored so that the course it follows may be reproduced.

Another output function I have been experimenting with is computerized music. With the aid of the H8-2 parallel board, I added a *digital-to-analog converter (DAC)*. This converter, together with a music generator circuit, produces different complex sounds to create such noises as gunshots, phaser sound effects, engine noises, and even steam engine sounds. I plan to do more work with electronic music, especially in the field of improving sound quality.

The Heath H8 is a good choice for the novice because of the simplicity involved in its assembly, its reasonable cost, and its expandability. At the end of the construction process, you will know why components are where they are and what their functions are. For someone like me, who enjoys electronics but has never had any exposure to computer electronics, the H8 is a valuable learning tool. ■

---

service department in Michigan resulted in the delivery of replacement parts to Concord NH two days later. In addition to excellent assembly instructions, the kit included special tools, all required solder, etc.

After building the H8 and acquiring the H9, Mark used the system to do homework and wrote programs to solve problems posed in school. His intimate knowledge of the hardware of the computer (gained from constructing it) was of great benefit, because he is interested in interfacing the computer with other devices. For example, initial add-on projects include a remote relay board for controlling the lights, heat, etc, in his home; a music board having three channels (for three-part harmony) extending over several octaves and permitting control of the tone and pitch; and various projects from "Ciarcia's Circuit Cellar" (a monthly feature in BYTE magazine).

While Mark's younger sister, Lisa, is not as interested in the hardware as Mark, she promptly began writing programs of her own to be run on the Heathkit system. His younger brother Ron also began playing computer games and started writing a football program of his own.

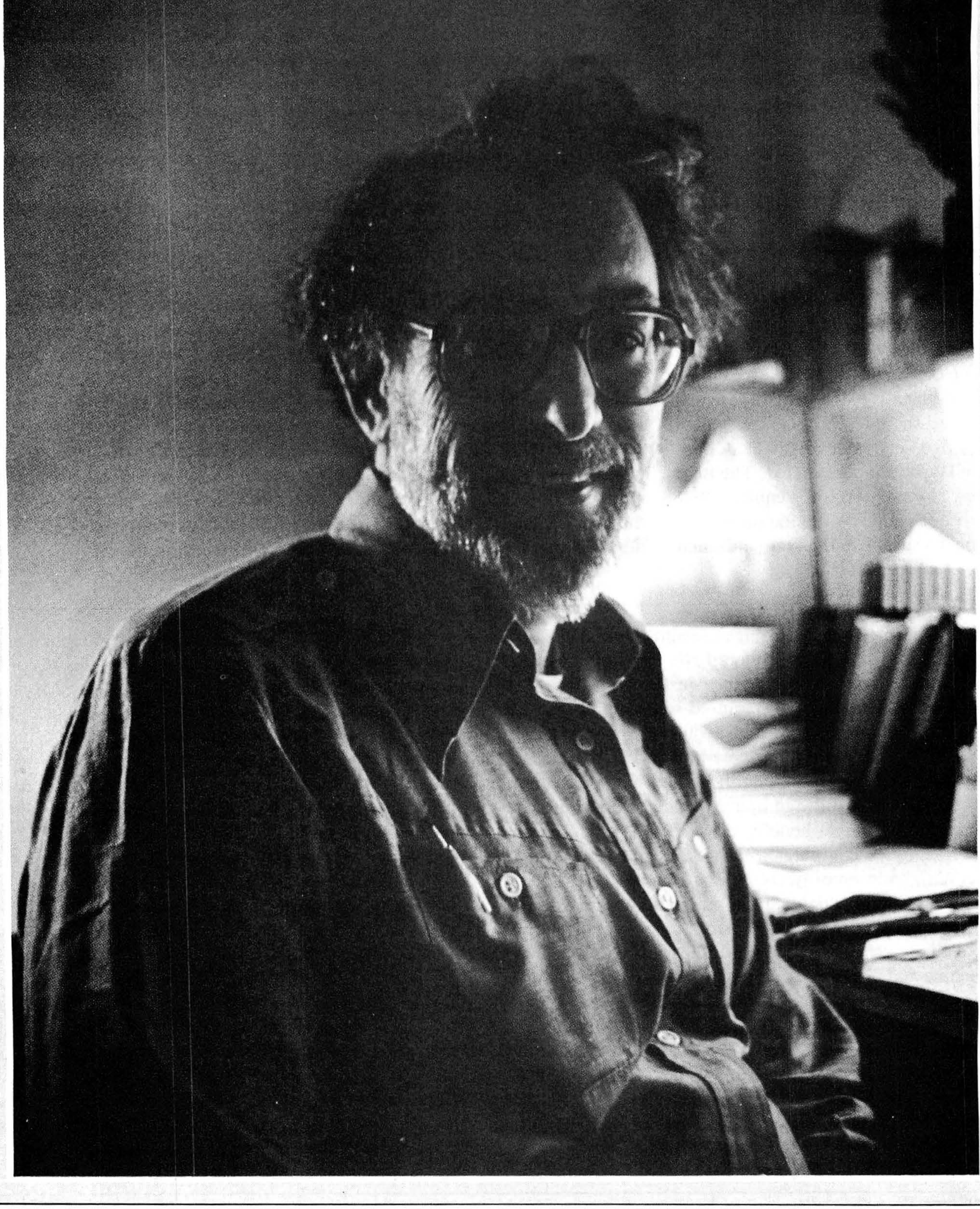
Mark may be getting help from Ork or beyond. One morning he found a page ripped from his notebook and a pen beside his bed. On the page was scrawled a computer program that turned out to be a game he calls "Gunner." Using this program, the computer places an object at a random distance and angle from the position of the person playing the game, and the gunner has five shots to try to blast the target. After entering the data concerning the angle of projection for firing each shot, the computer informs the player how many feet short or long the shot was relative to the target. The game is interesting enough that Heath is currently evaluating it.

Mysteriously, however, the program is not in Mark Snow's handwriting. He "didn't know the trigonometry used in writing the program . . . and . . . still [doesn't] know how to use most of the math in the program." Mark has no recollection of writing it. Although the handwriting is similar to his own, it is almost identical to that of a close friend of his . . . the friend was in Detroit the entire summer that the incident happened.

Even discounting the unusual events that Mark's computer has brought to his life, the computer has become a convenient, educational, and valuable tool for Mark Snow and the other Snows of Concord. ■

Profile of SOL LIBES

by Larry Barnes



Is Sol Libes a computer scientist who does an inordinate amount of writing, or is he a writer who has become an expert on computers? Actually, neither is the case, or, perhaps, both are true.

Sol will tell you that he's an educator. He teaches electronics and computer science at Union County Technical Institute, a junior college near his home in New Jersey. He admits to being a self-taught computer authority, but says that when he first became interested in the field there weren't many convenient places that offered a complete spread of courses on the subject.

He was, however, well prepared to learn the subject on his own. He had earned his BS in electrical engineering from City University of New York in 1953, and then spent 13 years as a technical writer and electrical engineer. His master's in electrical engineering from Rutgers University came in 1971.

A significant part of the computer fraternity is now learning from Sol Libes. He is the author of a dozen books on the art, plus several hundred magazine articles. He also writes regularly for BYTE magazine. It is difficult to pick up any computer publication without finding a significant and practical contribution by Libes. (See "How To Get Bargains in Used Computer Peripherals" in the Summer 1979 onComputing.)

Libes is well known as the founder and longtime president of the Amateur Computer Group of New Jersey (ACG-NJ), billed as the country's largest group, with membership approaching 1000. The club grew out of the local computer buffs' need for a forum in which to trade problems and solutions. Many of these people were Sol's students.

The word *amateur* in the title of the computer group is misleading. A breakdown of the membership indicates that well over half are professionally involved with computers, many as designers, programmers, technicians, or managers. Their main purpose in being members is to improve themselves professionally and to keep up with a rapidly advancing science.

But not all are professionals. Their youngest member is a 9-year-old, described by Libes as a "programming genius." They also have a 16-year-old, Tod Loofbourrow, who built a robot when he was 14, and has been improving it ever since. This robot is no ordinary mechanical man, but rather a working robot complete with ultrasonic sensors and a microprocessor brain.

Is Sol Libes a computer scientist who does an inordinate amount of writing, or is he a writer who has become an expert on computers? Actually, neither is the case, or perhaps, both are true.

Regular meetings are held monthly. They start with individual classes which are teaching sessions in various phases of the computer art. The general meeting follows, usually featuring a guest speaker of note. A flea market, in which members trade equipment and software, is a popular event at each meeting. Subgroups meet twice a month for the benefit of those members using specific types of equipment.

An important function of the organization is publication of their monthly newsletter, the "ACG-NJ News," written in good part by Sol Libes. His regular column, "Bits & Bytes," is an update on what is going on in the industry, from both a technical and a business viewpoint.

After holding the office of president of the ACG-NJ for 3½ years, Sol stepped down to become the corresponding secretary. He is listed on the masthead of the newsletter as assistant editor, contributing editor, and photographer. It was entirely proper, therefore, that in December 1978 the ACG-NJ awarded him a plaque, ". . . for his determination and diligent efforts in forming and running the largest computer club in the country."

Somehow Sol finds time to be a featured speaker at IEEE (Institute of Electrical and Electronics Engineers) seminars on computers, and to serve as a consultant to businesses contemplating the use of computers, or with specific computer problems.

There is a private Sol Libes who lives in a beautiful home in a suburban community in New Jersey. The home is equipped with a book-lined study, somewhat messy as all proper studies are, and an area in the basement where the word-processors and microcomputers live, frequently in various states of disassembly and incipient improvement.

Sol's wife, Lennie, is professor of mathematics at County College of Morris, a local junior college. Their daughter is at MIT pursuing a doctorate in chemistry, and their son is majoring in mathematics at Rutgers, where he is a junior.

Sol and Lennie are hikers, though where they find the time for it is a mystery. They've hiked through Peru, Egypt, and, by the time you read this, Mayan country, in pursuit of yet another interest, archeology.

Sol Libes and his family both inspire and dismay those of us who can never find enough time to do all of the necessary things that should be accomplished, and who like work so much that we always save some of it to be done next week.

We saw Sol at the National Computer Conference recently, where he was manning a booth for a friend. We inquired about his latest project.

"It has to do with personal computers," he said.

We weren't surprised. ■

COMPUTER STORE DIRECTORY

update

Last issue we published a list of personal computer stores, and asked readers to report any omissions or new additions to us.

We were inundated, as you can see!

Computer stores are literally springing up everywhere, and we are doing our best to keep up with them. Below is a list of all the stores that were left out of our last report. We apologize to everyone who was left out, and would appreciate hearing about additions or corrections to this list. Please address all correspondence to Store List, onComputing, 70 Main St, Peterborough NH 03458.

0·10000

Micro Computer Store of Puerto Rico
1568 Av Central
Capana Terrace
Rio Piedras Puerto Rico 00921

Computer Packages Unltd
244 W Boylston
West Boylston MA 01583

The Computer Store
120 Cambridge St
Burlington MA 01803

Computerland - Boston
214 Worcester St
Wellesley MA 02181

Worldwide Electronics
130 Northeastern Blvd
Nashua NH 03060

Computerland
80 Coburn Woods
Nashua NH 03060

Computerland - Hartford
57 Pratt St Rm 808
Hartford CT 06103

Computerland - Fairfield
1700 Post Rd
Heritage Sq
Fairfield CT 06430

Multi Business Computer Sys
Portland Professional Ctr
28 Marlborough St
Portland CT 06480

The Computer Place
21 Atlantic St
Stamford CT 06901

Computer Nook
Pine Brook Plz Rt 46
Pine Brook NJ 07058

Micro Business Systems
Pine Brook Plz Rt 46
Pine Brook NJ 07058

Typetronic Computer Store
806 Rt 17
Ramsey NJ 07446

Computerland - Bergen County
Hwy E65 Rt 4
Paramus NJ 07652

Stonehenge Computer Co
89 Summit Av
Summit NJ 07901

DAS Computer Supply Co
7870 Airport Hwy
Pennsauken NJ 08109

1·20000

Computer World
519 Boston Post Rd
Port Chester NY 10573

Computerland - Nassau County
79 Westbury Av
Carle Pl Long Island NY 11514

Audiotech Computer Shop
Rt 28
W Hurley NY 12491

ASD Computer Center
Rt 55 Van Wyck Plz
Poughkeepsie NY 12603

International Automation Inc
2nd floor Mellon Bank Building
New Kensington PA 15068

The Computer House
1000 Greentree Rd
Pittsburgh PA 15220

Scorpionics
249 Grandview Blvd
Butler PA 16001

Microcomputer Systems Inc
243 W Chocolate Av
Hershey PA 17033

Computerland - Lehigh Valley
1360 Armstrong Rd
Bethlehem PA 18017

P/MI Computers
605 Northern Blvd
Chinchilla PA 18410

A B Computers
115 E Stump Road
Montgomeryville PA 18936

Computerland - Paoli
81 E Lancaster Av
Paoli PA 19301

2·30000

Surplus Electronics
9600 Baltimore Blvd Rear
College Park MD 20740

Computer Bargains
1749 Rockville Pike
Rockville MD 20855

Computer Center of Columbia
9143 G Red Branch Rd
Columbia MD 21045

The Math Box
2627 University Blvd
Wheaton MD 21204

Computer Works
Rt 6 POB 65A
Harrisonburg VA 22801

Ohio Valley Electronic Systems Inc
190 Glenwood Rd
Wheeling WV 26003

Computer Corner
22 Beechurst Av
Morganton WV 26505

Corner Computer Store
900-902 Spring Garden St
Greensboro NC 27403

Computerroom
4231 Monroe Rd
Charlotte NC 28205

World of Computers
5849 Dorchester Rd
Charleston SC 29405

3·40000

Micro One
110 W Hancock Av
Athens GA 30601

Computerland - Jacksonville
2777 - 6 University Blvd W
Jacksonville FL 32217

Creative Computer Systems
1325 N Atlantic Av Suite 4
Cocoa Beach FL 32931

Computer Age
1308 North Federal Hwy
Pompano Beach FL 33062

Budget Computer Systems, Inc.
5111 S.W. 8th Street
Suite 103
Miami, FL 33134

Computer Village
931 SW 87th Av
Miami FL 33174

Computerland - Miami
3963 N Federal Hwy
Ft Lauderdale FL 33308

The Computer Store
21 Clearwater Mall
Clearwater FL 33516

Computer Center
6578 Central Av
St Petersburg FL 33707

Byte Shop - Huntsville
POB 5406
Huntsville AL 35805

Computerland - Huntsville
3020 University Dr NW
Huntsville AL 36805

Computerworld
339 Lakeside Dr
Ashland City TN 37015

Electronics World
3425 Denny Av
Pasca Goula MS 39567

4·50000

Computerland - Louisville
813 B Lyndon Ln
Louisville KY 40222

Astro Video Electronics
504 E Main St
Lancaster OH 43130

Computerland - Columbus
6429 Busch Rd
Columbus OH 43229

Computerland - Cleveland West
4579 Great Northern Blvd
North Olmsted OH 44070

Basic Computer Shop
2671 W Market St
Akron OH 44313

21st Century Shop
16 Convention Way
Cincinnati OH 45202

Forbes Microsystems Inc
35 N Broad St
Fairborn OH 45324

Micro Computer Center
7900 Paragon Rd
Dayton OH 45459

Computer Center
19819 Orchard St
South Bend IN 46637

Computerland - Indianapolis
410 E Main St
Washington IN 47501

Computer Connection
38437 Grand River
Farmington Hill MI 48018

Computer Mart of Royal Oak
Bywood Sq
560 W 14 Mile Rd
Clawson MI 48017

Computerland - Southfield
29763 Northwestern Hwy
Southfield MI 48034

Computerland - Rochester
301 S Livernois Av
Rochester MI 48063

American Computer Center
4132 Woodward Av
Royal Oak MI 48073

Computer Center
28251 Ford Rd
Garden City MI 48135

Computer House Division
1407 Clinton Rd
Jackson MI 49202

5·60000

Business Data Processing
4347 Merle Hay Rd
Des Moines IA 50310

Memory Bank
4128 Brady St
Davenport IA 52806

Zim Computers
5717 Xerxes Av No
Brooklyn Center MN 55429

Computerland - Minneapolis
8070 Morgan Circle Dr
Bloomington MN 55431

6·70000

Computerland - Mundelein
1500 S Lake St
Mundelein IL 60060

AAA Chicago Computer Center
120 Chestnut Ln
Wheeling IL 60090

Personal Computer of Chicago
59 E Lake St
Chicago IL 60601

Personal Computers
100 E Ohio
Chicago IL 60611

Computer Products
207 100F Bldg
Lincoln IL 62656

Computerland - St Louis
2355 Sarthe Ct
Maryland Hts MO 63043

Forsythe Computer
7748 Forsythe
Clayton MO 63105

Computer Country
4479 Lemay Ferry Rd
St Louis MO 63129

Computer Mart
1915 Noiand Rd
Independence MO 64055

Computers - ASP
7115 NW Barry Rd
Kansas City MO 64152

Computer Mart
1904 B East Meadowmere
Springfield MO 65804

Computerland - Independence
10049 Santa Fe Dr
Overland Park KS 66212

7·80000

Computer Place
3340 Highland Rd
Baton Rouge LA 70802

Computerland - Oklahoma City
10621 N May Av
Oklahoma City OK 73120

Compushop
POB 2067
Richardson TX 75080

K A Computer Store
9090 Stemmons Frwy
Dallas TX 75247

Microage Computer Store #6
1220 Melbourne Dr
Hurst TX 76053

The Computer Center
210 Fre Mar Valley
Wichita Falls TX 76301

Southwest Technical Products Co
219 W Rhapsody
San Antonio TX 78216

Computer Corner Inc
1800 S Georgia #4
Amarillo TX 79109

Computer Technology Associates
300 Thunderbird
El Paso TX 79912

8·90000

Computerland - North Denver
9321 W 92nd Av
Westminster CO 80020

Miti Mini Computer Co
8535 E Orchard Rd #401
Englewood CA 80110

Neighborhood Computer Store
13045 W Alameda Av
Lakewood CO 80228

Computer Shack
3120 San Mateo
Albuquerque NM 81110

R & L Data Systems Inc
684 Shoup Av
Idaho Falls ID 83401

Byte Shop
418 N Orchard
Boise ID 83704

Northwest Computer Center
6457 Fairview Av
Boise ID 83804

Computerland - Phoenix
3152 E Camelback Rd
Phoenix AZ 85016

Microage Computer Store #8
2525 N Scottsdale Rd
Scottsdale AZ 85257

Byte Shop of Arizona
1425 W 12th Pl #101
Tempe AZ 85281

Sunrise Computer Mart and Data Service
1437-A Wyoming Blvd NE
Albuquerque NM 87110

90000

Byte Shop #74
11611 San Vincente Blvd
Los Angeles CA 90049

Micronomic Computer Systems
10871 W Pico Blvd
Los Angeles CA 90064

DES Data Equipment Supply
8315 Firestone Blvd
Downey CA 90241

Computer Company of South Bay
15818 Hawthorne Blvd
Lawndale CA 90260

Computerland - South Bay
16720 S Hawthorne Blvd
Lawndale CA 90260

Computer Place
2530 W Sepulveda Blvd
Torrance CA 90505

Computer Forum
14052 E Firestone Blvd
Santa Fe Spring CA 90670

Byte Shop - Cerritos
11470 South St
Cerritos CA 90701

Computers Unltd
2262 Pacific Coast Hwy
Lomita CA 90717

Computerland - Pasadena
81 N Lake St
Pasadena CA 91101

Computer Components
3808 W Verdugo
Burbank CA 91505

Byte Shop
8038 Clairmont Av
San Diego CA 92111

Computerland - San Diego East
2992 Navajo Rd
El Cajon CA 92020

Computer Age
4688 Convoy St
Suite 105
San Diego CA 92111

The Computer Merchant
5107 El Cajon Blvd
San Diego CA 92115

Computerland - Tustin
104 W First St
Tustin CA 92680

Computer City
3941 S Bristol St
Santa Ana CA 92704

Computer World
6791 Westminster Av
Westminster CA 92683

The Computer Warehouse
4100 Easton Dr
Suite 11
Bakersfield CA 93309

Computerland - Santa Maria
223 S Broadway
Santa Maria CA 93454

Computer Place
2638-4 Carmel Rancho Ln
Carmel CA 93923

Computer Plus
1324 S Mary Av
Sunnyvale CA 94087

Computerland - San Francisco West
117 Fremont St
San Francisco CA 94105

Byteland
38256G Glenmoor Dr
Fremont CA 94536

Computerland - Hayward
22634 Foothill Blvd
Hayward CA 94542

Computer Store of San Leandro
701 MacArthur Blvd
San Leandro CA 94577

Microsun Computer Center
2989 N Main St
Walnut Creek CA 94596

Computerland - Matin
1930 4th St
San Rafael CA 94901

California Business Computers
281 Jaggers
San Jose CA 95119

Grass Valley Computer System
POB 678
Rough and Ready CA 95975

Computer Emporium
Galleria Main Floor
921 SW Morrison
Portland OR 97205

Computer Solutions
531 SW 4th St
Corvallis OR 97330

Custom Computer Corp
K-Mart Plz
259-2 Barnett Rd
Medford OR 97501

Byte Shop
20101 44th W
Lynnwood WA 98036

Micro Business Systems
6401 Roosevelt Way NE
Seattle WA 98115

Byte Shop
2605 2nd Av
Seattle Wa 98121

Computerland - Sacramento
1100 Howe Av Apt 329
POB 25708
Sacramento CA 98525

FOREIGN

Creative Computers
208 - 1595 McKenzie
Victoria BC V8N 1R4 Canada

Computerland - Toronto
2180 Yonge St
Toronto Ontario M4S 2B9
Canada

Computerland - Winnipeg
715 Portage Av
Winnipeg Manitoba R3G
OM8 Canada

Compushop Canada Ltd
103 - 723 14th St NW
Calgary Alberta Canada T2N2A4

Computerland - Calgary
180 White Glen Cres NE
Calgary Alberta T1Y 2Z5
Canada

Computelex
16 Rue de Lancry
75010 Paris France

Computerland - Paris
44 Elysee 2
78170 La Celle Rue
Cloud France

Computerland - Stockholm
Sjohagsuagon 19
12532 Stockholm Sweden

Computerland - Brussels
Square La Rousse 5
1060 Bruxelles Belgium

Byte Computer Shop
Favoritenstrabe 20
A - 1040 Vienna Austria

Computerland - Copenhagen
Brogade 9
3730 Nexo Denmark

Computer Components
14 Station Rd New Barnet
Herts UK ENS IQW
England

Personal Computer World
62 A Westbourne Grove
London England W2

Computer Store
POB 31 - 261
Aukland 9 New Zealand

Cochrane Computer World
c/o Emery Airfreight
Kalammunda POB 259
W Australia

Computerland - Melbourne
555 Collins St
Melbourne 3000 Australia

Computerland - Perth
City Mutual Pl
197 St Geroges Terrace
Perth W Australia 9000

Computerland - Sydney
55 Clarence St
Sydney NSW 2000

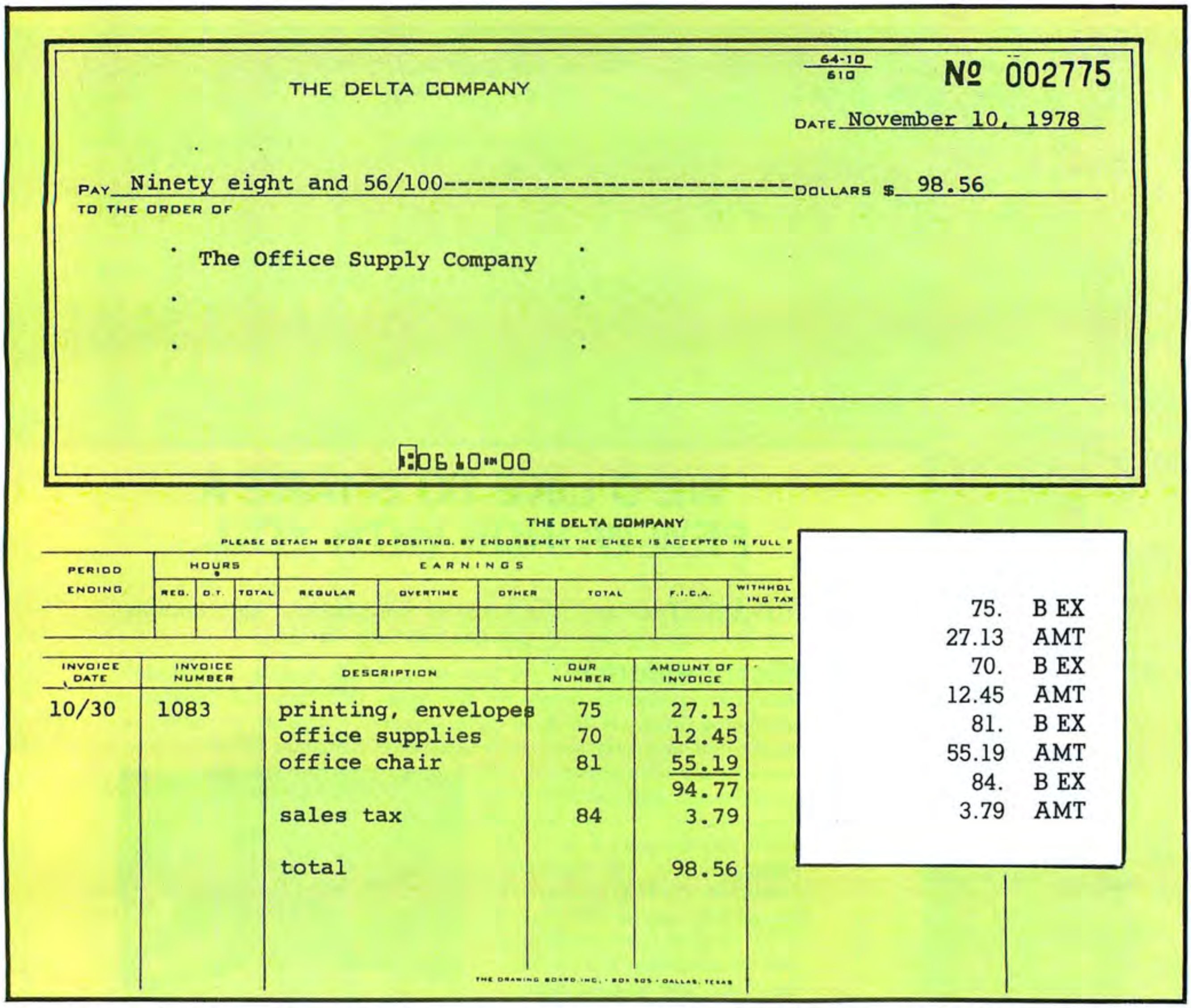
Computerland - Brisbane
127 Creek St
Brisbane Australia TLX 41156

Integrated Computer Systems
STE 118 LIMKBLD Ortigas Av
San Juan Metro Manila
3134 Philippines

Computerland - Philippines
Ground Floor, Alexander
House, 132 Amorsolo St
St Legaspi Village Makati
Metro Manila 3117

Computer Center PTE Ltd
5345 Woh Hup Complex
Beach Rd 7 Singapore

Computerland - Singapore
1101 11th floor Fook Hai
Bldg
150 South Bridge
Singapore 1


THE DELTA COMPANY

64-10 / 610 № 002775

DATE November 10, 1978

PAY Ninety eight and 56/100------------------------- DOLLARS $ 98.56

TO THE ORDER OF

The Office Supply Company

⑆0610⑈00

THE DELTA COMPANY

PLEASE DETACH BEFORE DEPOSITING. BY ENDORSEMENT THE CHECK IS ACCEPTED IN FULL P

PERIOD ENDING	HOURS REG.	O.T.	TOTAL	EARNINGS REGULAR	OVERTIME	OTHER	TOTAL	F.I.C.A.	WITHHOLDING TAX

INVOICE DATE	INVOICE NUMBER	DESCRIPTION	OUR NUMBER	AMOUNT OF INVOICE
10/30	1083	printing, envelopes	75	27.13
		office supplies	70	12.45
		office chair	81	55.19
				94.77
		sales tax	84	3.79
		total		98.56

THE DRAWING BOARD, INC. · BOX 505 · DALLAS, TEXAS

75.	B EX
27.13	AMT
70.	B EX
12.45	AMT
81.	B EX
55.19	AMT
84.	B EX
3.79	AMT

Figure 2: Voucher check shows one way to record business expenses. Inset illustrates distribution on printed tape.

Text continued from page 14:

do if you need all 100 registers. But if sixty registers are all you need, buy a calculator/printer, assign numbers to your expense categories (00-59), and you're ready to go right away. There is an option here: If you don't need this many counters, you can use the TI-58 calculator, which has thirty registers ready to go, and sixty when you repartition the memory. The TI-58 costs about $100, and works with a printer also. It looks like the TI-59, but it has fewer registers, and you can't use it easily with a control program.

Optional step 2b — Buy a calculator and use a control program. Although using a calculator as it comes off the shelf can get you going right away, adding a control program simplifies the procedure to A-B-C steps, and almost anyone can do your expense analysis with only a few minutes of instruction.

What is a control program? It's a list of procedural steps recorded on a small magnetic card. This magnetic card tells the calculator and the automatic printer what to print and when to print it, how many counters to use, when to print a final total, and similar procedural steps. (See the boxed program).

Here's the way a control program I've set up works: There are buttons labeled A, B, C, and D across the top of the calculator. These buttons tell the calculator what information you're punching on the keyboard. When you press one of these buttons, the calculator and its program take over and do or print the right thing. (See figure 4.)

It's not actually necessary for you to remember what each of these keys is for. The purpose of each key is written on a special tab above the lettered keys; this tab is shown in the photo above the top row of keys.

Examples of the printed record resulting from the control program are shown in figure 5. With the TI-59 calculator, the program is recorded on a magnetic strip and run into the calculator whenever you start an expense distribution. You may have someone prepare a control program for you locally, or you can try the control program in this article. The store where you buy your calculator may know several programmers. Show this article to them and explain what you want. If

SUMMARY
11.
12365.87
14.
21587.21
15.
19741.96
16.
24753.51
19.
25852.96
20.
3300.00
23.
512.52
25.
3985.41
34.
500.00
44.
412.69
48.
751.52
55.
1205.63
57.
1847.63
57.
1847.63
62.
8896.74
65.
2210.00
68.
526.25
70.
2005.85
74.
741.85
75.
516.86
76.
21045.67
78.
451.87
82.
831.71
84.
1506.23
85.
614.23
90.
745.81
92.
687.71
95.
341.67
158939.36

Figure 3: Summary totals of all expenses accumulated in calculator registers. Account numbers correspond to those shown as examples in figure 1.

Figure 4: Using the control program. First enter the program into the TI-59 calculator (see figure 6, page 76). Buttons A, B, C, and D can then be used to perform the four functions listed below. In actual use, the data is entered first, then the appropriate button is pressed.

Step	Procedure	Then Punch Button	Calculator Does This
A	Punch in amount of expense to distribute (example $327.14)	**A** (remember **A** is for **A**mount)	Prints the expense amount on the tape.
B	Punch in the expense category number (25 for example)	**B** (remember **B** is for **B**usiness expense.	Prints the expense category number on the tape; adds the dollar amount from step A to register 25, and to a grand total register.
C	Punch in the check number for identification (this is an optional step; you may omit it if you want)	**C** (remember **C** is for **C**heck number)	Prints the check number for identification. You can enter steps A, B and C in any sequence, or skip C altogether.
D	Continue entering expenses and their expense category numbers as described above, until you're done with all expense items for the month, quarter, or year.	**D** (remember **D** is for **D**one)	Prints a final total of all amounts you've keyed in, and then automatically prints the contents of each expense register, with the register number corresponding to your expense categories.

1234.	CKND
11.	B EX
215.63	AMT
61.	B EX
115.92	AMT
15.	B EX
786.35	AMT
1235.	CKND
61.	B EX
2162.85	AMT
1236.	CKND
15.	B EX
122.50	AMT
3403.25	TOT

Figure 5: Example of printed tape when control program is used. At the end of all expense entries, the calculator prints the total of all dollar amounts entered.

you're unable to obtain assistance locally, write to me in care of onComputing and I'll try to assist.

The cost of the printer is about $175 at this writing. Your time savings should pay for the equipment cost, and you'll likely find numerous other uses for the calculator/printer combination.

To summarize what you need to do, here are the steps:

(1) Buy the calculator and printer you decide to use.

(2) Decide whether you'll use the *off-the-shelf* or *program control* method.

(3) Assign 2-digit numbers to your expense categories.

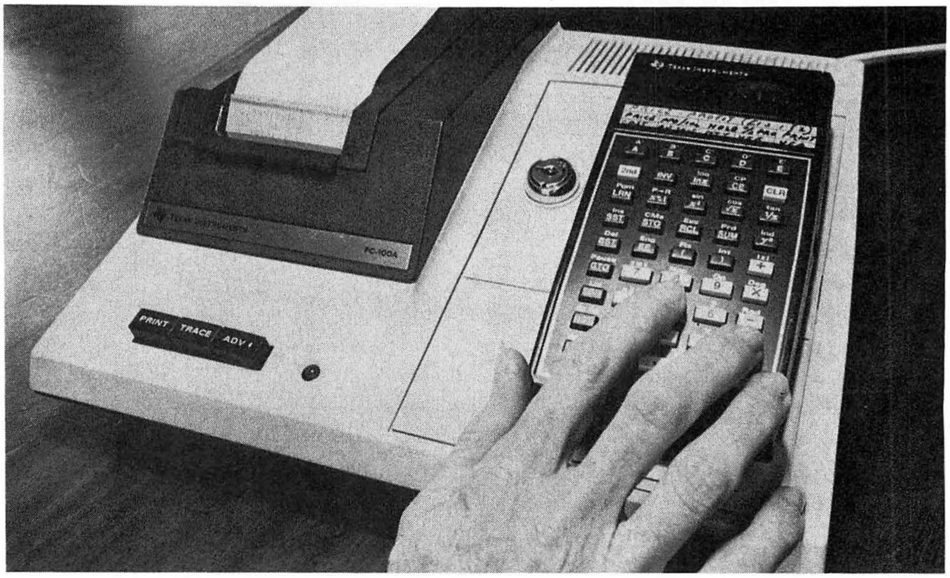
(4) Get a program for your calculator if you use this method, or use the program described herein.

Figure 6: The control program written for the Texas Instruments TI-59 programmable calculator. See figure 4 for details about using the program.

000	43	RCL
001	05	05
002	69	OP
003	04	04
004	25	CLR
005	91	R/S
006	76	LBL
007	12	B
008	42	STO
009	02	02
010	87	IFF
011	01	01
012	99	PRT
013	86	STF
014	02	02
015	91	R/S
016	76	LBL
017	11	A
018	42	STO
019	01	01
020	87	IFF
021	02	02
022	99	PRT
023	86	STF
024	01	01
025	91	R/S
026	76	LBL
027	13	C
028	69	OP
029	06	06
030	91	R/S
031	76	LBL
032	99	PRT
033	43	RCL
034	04	04
035	69	OP
036	04	04
037	43	RCL
038	02	02
039	69	OP
040	06	06
041	43	RCL
043	69	OP
044	04	04
045	43	RCL
046	01	01
047	44	SUM
048	00	00
049	74	SM*
050	02	02
051	71	SBR
052	44	SUM
053	81	RST
054	76	LBL
055	14	D
056	43	RCL
057	08	08
058	42	STO
059	02	02
060	43	RCL
061	07	07
062	42	STO
063	01	01
064	98	ADV
065	43	RCL
066	06	06
067	69	OP
068	04	04
069	43	RCL
070	00	00
071	71	SBR
072	44	SUM
073	98	ADV
074	25	CLR
075	42	STO
076	00	00
077	69	OP
078	00	00
079	43	RCL
080	10	10
081	69	OP
082	02	02
083	43	RCL
084	09	09
085	69	OP
086	03	03
087	69	OP
088	05	05
089	76	LBL
090	90	LST
091	73	RC*
092	07	07
093	22	INV
094	67	EQ
095	96	WRT
096	76	LBL
097	97	DSZ
098	69	OP
099	27	27
100	97	DSZ
101	08	08
102	90	LST
103	43	RCL
104	02	02
105	42	STO
106	08	08
107	43	RCL
108	01	01
109	42	STO
110	07	07
111	43	RCL
112	06	06
113	69	OP
114	04	04
115	43	RCL
116	00	00
117	98	ADV
118	71	SUM
119	44	SUM
120	98	ADV
121	98	ADV
122	98	ADV
123	42	STO
124	00	00
125	81	RST
126	76	LBL
127	96	WRT
128	43	RCL
129	07	07
130	99	PRT
131	73	RC*
132	07	07
133	44	SUM
134	00	00
135	71	SBR
136	44	SUM
137	25	CLR
138	72	ST*
139	07	07
140	61	GTO
141	97	DSZ
142	76	LBL
143	44	SUM
144	58	FIX
145	02	02
146	69	OP
147	06	06
148	22	INV
149	58	FIX
150	92	RTN
151	00	0
152	00	0
153	00	0
154	00	0
155	00	0
156	00	0

007	12	B
017	11	A
027	13	C
032	99	PRT
055	14	D
090	90	LST
097	97	DSZ
127	96	WRT
143	44	SUM

.0	00
0.	01
.0.	02
13303700.	03
14001744.	04
15263132.	05
37323700.	06
11.	07
89.	08
3030133545.	09
3641.	10
0.	11
0.	12
0.	13
0.	14 ■



The TI-59 calculator with PC-100A printer.

Text continued from page 25:

statements or commands are further used to carry out specific operations. A typical example of a FORTRAN expression is shown below.

Symbol	Meaning
+	addition
–	subtraction
*	multiplication
/	division
**	exponentiation

In FORTRAN,

The equation: $X = \dfrac{-B + \sqrt{B^2 - 4AC}}{2A}$

becomes

X = (–B + (B**2 – 4*A*C)** .5)/(2*A)
(NOTE: Raising an expression to the .5 power is the same as square root.)

FORTRAN is a compiler language. Once a programmer prepares the program, it must be entered into the computer, compiled into object code, then executed.

BASIC

The higher-level language most widely used with microcomputers is BASIC, which stands for *beginners all-purpose symbolic instruction code*. Like FORTRAN, BASIC uses English-like statements and algebraic expressions to state a problem. BASIC was developed at Dartmouth College in the 1960s, primarily to teach beginners the fundamentals of computer languages and programming. However, BASIC has been widely implemented, not only on large-scale computers, but also on microcomputers. It has become the single most popular microcomputer language. Virtually every available personal computer and microcomputer uses the BASIC language. BASIC was not optimized for any particular class of applications problem. It can be used for a variety of applications, from mathematical problem solving, to business data processing, to computer-aided instruction.

While BASIC compilers do exist, most BASICs are interpreters. In fact, most personal computers come with a built-in BASIC interpreter. When such computers are turned on, they are ready for immediate use. A user can sit down and begin entering programs in the BASIC language. The interactive nature of this language allows the user to get immediate effectiveness from the machine. BASIC compilers are available for some computers in which demanding applications require shorter programs that execute faster.

Pascal

Another computer language that is gaining rapid popularity is Pascal. Developed by a Swiss professor,

Nicklaus Wirth, Pascal is designed to be easy to teach. Several years ago this relatively obscure language was discovered to be extremely easy to learn and use. It is also a powerful language that not only can be used to solve a wide variety of applications problems, but can also be effectively used to write higher-level languages and systems programs. Compilers and interpreters are usually written in assembly language for their host machine. Now such languages can be more easily written in Pascal. The use of Pascal is growing daily since it has been implemented on many microcomputers.

The main benefits of Pascal are its efficiency and portability. Its efficiency helps to cut programming time and programming errors. Programmers become more productive with Pascal. Its portability makes it usable on a wide range of machines. A program written in Pascal on one computer can be run on another computer. This is not always possible with other languages like FORTRAN and BASIC because of the many different "dialects" that exist. This portability comes from Pascal's unique structure. Pascal is a compiler. It converts the user's source program into an intermediate *pseudo* language, called *p-code*, rather than into machine code. An interpreter written in machine code for the host computer then executes the p-code formatted user's program (see figure 6). The p-code is standardized, but the interpreter is customized to the specific computer. The interpreter is easy to write, therefore Pascal can be quickly implemented on almost any computer. A Pascal program can then be transported to any machine with an interpreter.

Systems Utilities

The other category of systems software is often called systems utilities. These are a wide variety of general-purpose programs that greatly speed up and simplify the use of the computer. They are written to enhance man-to-machine communications and improve programming efficiency. Most microcomputers have the following types of systems programs: *editor, monitor,* and *operating system*. Let us consider these most commonly used systems programs.

Editor

Perhaps the most often used systems program is the editor. Editors are interactive programs that are stored in memory and allow the user to write a program,

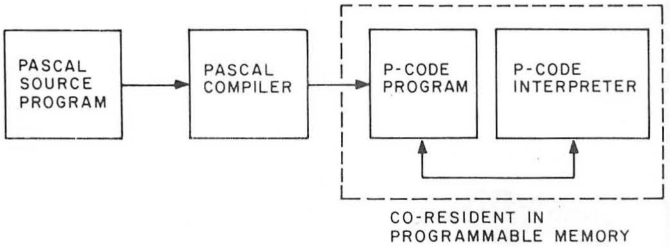


Figure 6: How a Pascal program is executed.

generate text, or make a wide variety of changes and additions to either of these. The editor is a systems program that is typically stored on some mass, external media. When it is needed, it is called into memory. The programmer then communicates with the editor through the video or hard-copy terminal. Like an interpreter, the editor is interactive. All source programs for assemblers and compilers are typically written with the editor.

Since the editor can be used for creating any type of written material, it is naturally useful for writing letters or preparing manuscripts and reports. It can be used for many other word processing applications. Once a program or a block of text is created, it can then be edited as required. Lines, words, or single characters can be added, deleted, or otherwise modified. Once the final program or text is complete and error-free, the editor stores the final text in a file in programmable memory. The final text can also be stored on external tape or disk, or printed out.

Monitor

Another frequently used systems program is the monitor. Almost all microcomputers use some form of monitor. The monitor is typically stored in read-only memory.

Some of the functions that can be performed by a monitor through the video or hard-copy terminal are listed below:

1. Enter instructions and data.
2. Display contents of memory locations or registers.
3. Change contents of memory or registers.
4. Execute programs.
5. Load and store program from and to external media.

All of these functions are usually performed in machine language. Most monitors use either octal or hexadecimal designations when entering or displaying memory or register information. Addresses are also expressed in octal or hexadecimal notation. As you can see, the monitor is a program that essentially permits convenient entry, display, debugging (ie: finding and correcting errors), and execution of machine-language programs.

Operating System

Operating systems are perhaps the most complex of all systems programs. Yet they are widely used with microcomputer systems because they simplify the operation of the computer and the development of programs tremendously.

An operating system is a set of programs that automates the management of all of the computing resources available in a computer system including the microprocessor, all peripherals, and software. This integrated collection of programs provides more efficient computer operation. Improved operating convenience is another benefit of an operating system.

The operating system is probably the most powerful and important piece of software in a computer system.

Operating systems are typically used only in larger microcomputer systems. They require some mass storage media such as magnetic tape or floppy disk. Most microcomputer systems use operating systems only with floppy disk mass storage. For that reason, these operating systems are referred to as disk operating systems (DOS).

Figure 7 shows one way to visualize the operating system (OS). The OS effectively isolates the hardware from the user. The user communicates only with the OS, the various languages, and systems and applications programs that are available. The OS fields all requests for services, and totally eliminates the need for the user to deal with the hardware. The OS allows the user to concentrate on this application and prevents the programmer from having to spend a lot of time determining how things are to be specifically accomplished by the hardware.

An operating system has one major function: to manage the resources of the computer system. It allocates the resources on the basis of user need and systems capability. The three major functions of an operating system are:

1. File and software management.
2. I/O and peripherals management.
3. Memory management.

When a microcomputer is first turned on, the operating system program is transferred from floppy disk into the memory. Some simpler operating systems can be stored in read-only memory. With the operating system in memory, various system commands may be given by the user to perform a wide variety of operations. Let us consider the file and software management function.

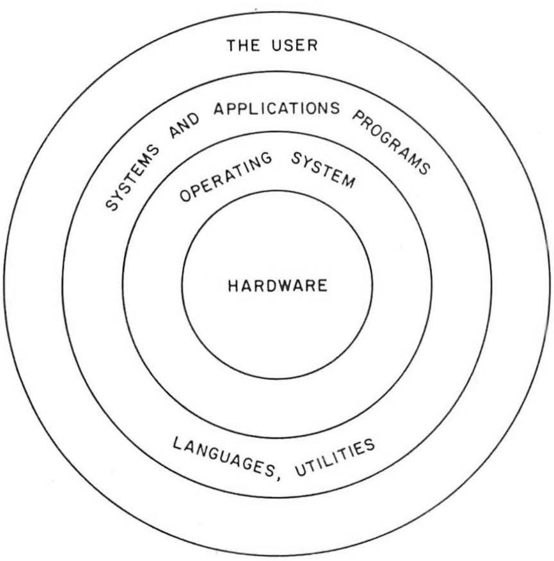


Figure 7: Visualizing the operating system.

The operating system supports a large library of typical user programs and files. The user need only tell the operating system which program is to be used. The operating system will retrieve it from the disk and bring it into main memory for the operator to use. These programs include the editor, the assembler, and a variety of higher-level languages. Many other programs and subroutines are available or can be created and stored in the library, including applications programs. Files of data can also be created, stored, and retrieved under this system.

I/O and peripherals management is another operating systems function. Operating systems also contain a variety of subroutines as peripheral device handlers and interrupt servicing routines. When writing programs, the user finds that most of the time and drudgery will be spent in developing the I/O routines for talking to the various peripheral devices. Much of any given program is concerned with inputting and outputting data. The operating system makes this easy by providing for all of these routines.

Memory management is another function of the operating system. The programs in an operating system are capable of determining how much usable programmable memory is available to a microcomputer. The operating system decides how this memory is to be used. For example, it will determine where in the memory a systems or applications program will be placed. The transferring or swapping of programs and data into and out of memory takes place automatically under the control of the operating system. Users don't have to concern themselves with this because the operating system does it automatically.

The operating system manages the available empty storage space on one or more external floppy disks. Should the user create a program that is larger than the available memory, the operating system automatically steps in and begins saving portions of the program on the external floppy disk, in order not to lose it. The operating system also decides where on the disk a user-created program will be stored. Typically we say that such memory and floppy disk operations are "transparent" to the user.

The operating system is probably the most powerful and important piece of software in a computer system. Along with the hardware, the operating system and other software form a complete, versatile, efficient, and easy to use system. ■

Letters

OnComputing welcomes correspondence from readers. Letters should be addressed to OnComputing, 70 Main Street, Peterborough, NH 03458.

Kudos for the onComputing Premier Issue (Summer 1979)

Dear Editor:

I should like to begin this letter by complimenting you and your staff on the excellent premier issue of onComputing. I was particularly impressed with the Theodore Cohen article "The Information Age." I feel that onComputing will not only serve to inform average personal computer hobbyists of the value and uses of their computer, but will also serve to inform the general public about the advantages of owning a personal computer.

John Stuart Koehler

Dear Editor:

Found your first issue very interesting, and hope that you keep up the good work. I particularly liked the product reviews. I presently own a TRS-80 Level 2 personal computer with 16 K bytes of memory.

Gerald B Clark
Newport ME

Dear Editor:

Congratulations on a very interesting first issue. I am sure onComputing will be very successful, especially if it continues to attract readers with such an edifying selection of articles. I have just recently become interested in a personal computer for the future, and found several of the articles both interesting and useful.

Richard A Polunsky
Austin TX

Dear Editor:

I recently received my first issue of onComputing, and I am very pleased with the magazine. I presently own an Ohio Scientific Challenger 1P personal computer.

Eric Sievers
Boulder CO

Dear Editor:

I just received your first issue of onComputing. One word to describe it: WOW! If there is a computing publication I do not subscribe to, it is only because I have not discovered it yet. Yours tops them all. Keep up the good work

Shirley R Jones
Winston Salem NC

PS: I do not believe I have ever read every article in a magazine until I read onComputing.

Dear Editor:

Congratulations on your first issue of what looks like a fine magazine. It is one much needed in the personal computing field. It fills a real need in helping me keep up-to-date on the personal computer systems available in the marketplace. Because of my job as an analyst/programmer, and my interest in microcomputers, I am frequently asked to give tutorials on computing, and to recommend small systems for personal use. Along these lines, I hope you will occasionally publish system updates of previously reviewed systems to let us know about the "latest-and-greatest" features.

Being an Apple II owner, I was interested in the review of that product. While agreeing with most of what Mr Cote said, I felt that he understated one of the strongest features of the Apple system — graphics.

The Apple has two graphics modes. The low resolution mode, with 16 colors, offers either a 40 by 48 grid or a 40 by 40 grid with 4 lines of scrollable text at the bottom of the screen. The high resolution mode, with 4 colors, offers either a 280 by 192 grid or a 280 by 160 grid with 4 lines of scrollable text at the bottom of the screen. The graphics is limited in that text and graphics cannot be mixed in the upper portion of the screen. (This limitation can be overcome with firmware available from Eclectic Corp.)

In both modes, images are created by plotting the points or lines describing them. The low resolution plotting commands are built into both versions of BASIC. The high resolution plotting commands are built into the Extended BASIC, but must be referenced by CALLs with the Integer BASIC. [A more detailed discussion of the Apple II system can be found in the May 1977 and March 1978 issues of BYTE magazine—ed]

*Mr Cote indicated his dislike of the position of the RESET key. Granted that it is in a bad location, the available documentation (*Apple II BASIC Programming Manual, Disk Operating System Manual*) explains how to recover from an accidental reset without losing the contents of memory.*

While the documentation is generally very good, the more serious user who wants to work with assembler routines and make use of the monitor utilities will find that it is up to the user to determine the location and use of these routines from the assembler listing of the monitor. Some of the information is now being published by the user groups.

Jim Fleming
Columbus IN

Ray Cote replies:

Although the documentation does state ways to recover from most accidental resets, there are still situations where recovery is not simple or possible. One such case is initializing the system in integer BASIC and then performing a software switch to floating-point BASIC contained on a plug-in card. If reset is pressed, the normal recovery procedure will result in the system trying to run in integer BASIC with the floating-point program destroyed.

Although there may be ways to get around this problem, I feel that a personal computer user (not someone who is necessarily concerned with knowing

the inner workings of the machine, but someone who just wants to use the computer) should not have to worry about such details. These problems should be taken care of by the available software. Apple has taken great strides toward solving this inconvenience with the advent of the Apple II Plus personal computer. When the reset button is pressed on the Apple II Plus, the system is reset to floating-point BASIC and the program in memory is not lost. However, I do not know what happens if you are using integer BASIC when reset is hit, since I have not had access to an Apple II Plus machine.

Magic for Your Micro

Dear Editor:

I have enjoyed the first two issues of your magazine very much. Special thanks go to Chris Morgan for revealing his magic trick in the Fall 1979 issue.

Keep up the good work.

David Rich
Santa Cruz CA

Dear Editor:

The "Magic for Your Micro" article in the fall 1979 issue was quite interesting as well as devious.

However, the program will not work on the Commodore PET, since the PET will delete any unnecessary spaces in the user's input. Thus the inputs used in the program will not work.

The input statements should be changed as follows:

From: 210 input "YOUR CARD IS BLACK"; A$

To: 210 print "YOUR CARD IS BLACK"; GO SUB 2000

The following subroutine should be added:

```
2000  A$ = " ":X$ = " ":POKE 525,0
2010  PRINT"[RVS ON] [SPACE]
      [CURSOR LEFT]"; : FOR X = 1
      TO 240: NEXT
2020  PRINT"[RVS OFF] [SPACE]
      [CURSOR LEFT]"; : FOR X = 1
      TO 230:NEXT
2030  GET X$: IF X$ = " "THEN 2010
2040  IF X$ = " "THEN 2070
2050  IF ASC(X$) = 13 THEN
      PRINT:RETURN
2060  PRINT X$;
2070  A$ = A$ + X$
2080  GOTO 2010
```

This subroutine not only prints a false cursor, but when the extra space is inputed, the cursor stays in the same place: ie, it doesn't move over one space to the right, making detection of the trick difficult.

Your magazine looks promising, and I'll look forward to future issues.

Tom Thiel
Covina CA

Thanks to David Rich, Tom Thiel, and the other readers who wrote in to comment on the magic program, including Walter I Hess, who rewrote the program in Pascal and suggested that readers with the ability to turn off the cursors on their displays could thus make the typing of the secret space that much more difficult to detect...ed

Product Reviews

Dear Editor:

Congratulations on your premier issue of onComputing. Your magazine was most interesting and kept my attention from cover to cover. I am really glad to see someone has finally got it all together in the form of a magazine devoted to micros.

Several of your articles were most informative. The four user reviews of the leading microprocessors for personal use provided useful information in some cases. I am sure many of your readers would agree that a more useful review could be provided if it were in the format of a consumer report where a committee evaluates a predetermined number of features and options for each of the systems being reviewed. Perhaps we can have an article like this in the future.

In the meantime keep up the good work and good luck with your new magazine.

Art Frenz
Sarasota FL

Writing With a Microprocessor

Dear Editor:

I have just finished reading your article "Writing With a Microcomputer" [onComputing, summer 1979, page 12] and found it thoroughly enjoyable.

As a research medical officer engaged in the development of specialized biomedical instrumentation since the late 1950s, I have been professionally involved with the design and application of computers since about 1966. This has included the use of assembly language, FORTRAN, BASIC, etc, as well as the development of special languages for simulation purposes.

Although I learned touch-typing (self-taught) while I was in school, a combination of consistent bad spelling and very frequent finger misadventures has contributed to an already well-developed abhorrence for paperwork.

The possibilities of using a computer as a typing aid were immediately obvious, and I have done so for all my correspondence and official paperwork since the late 1960s. I frequently wondered why nobody else seemed to be doing this, until just a few years ago when I began hearing some rumblings about the use of computers for typesetting by some of the printing houses. More recently, the expression "word processor" has appeared as editors and publishers have cried out about the paper deluge.

I imagine the main reason computer assisted paper shuffling has been so slow in developing has been the cost. Now anyone who can afford the price of a car can have a fairly respectable computer for his very own. Eventually everyone who does any writing will soon realize that the ordinary typewriter will go the way of hand laundry, slide rule and wooden pencil.

George Potor Jr MD
Fairborn OH

Error Messages

Dear Mr Morgan:

Ken Knecht's review of the Diablo printer in the Fall 1979 onComputing is interesting, accurate, and well-written.

There is one comment I would like to make to Mr Knecht and your readers regarding printwheel usage. He states that he has not yet tried a metal printwheel on his Diablo 1620. Please don't! Diablo's two basic printer mechanisms, plastic printwheel and metal printwheel, differ in several important respects. Hammer energy levels are much higher in the metal-wheel machines. From this standpoint, no damage to the machine or wheel would occur if a metal wheel were placed on a 1620, but print density would suffer. On the other hand, use of a plastic printwheel on a metal-wheel machine (such as the Diablo 1650) would cause rapid wear out of the plastic wheel.

Another factor is that the two machine types have different printwheel servo systems. Each is optimized for the inertial characteristics of the wheel type that it is designed to drive. Interchanging wheel types would have a detrimental effect on print quality.

Stanley Silverman
OEM Product Marketing Manager
Diablo Systems Inc
Hayward CA

Mysterious Magic

Several readers wrote to take me to task for saying that my magic trick in BASIC (see onComputing, Fall 1979, page 19) will run on the TRS-80 as written. (See Letters, page 81, for a report about running the program on the Commodore PET computer.) I must confess that I was too hasty in claiming that the program could do this.

Also, problems arose from the fact that the program as originally written was much longer and more complicated than it appears in the article. I removed an extensive section devoted to color graphics effects in order to streamline the program and make it as universal as possible. However, I neglected to remove some extraneous lines from it. As far as making the program run on the TRS-80, I quote from reader H C Smartt of Eugene OR:

> *"When I first ran into trouble with lines 110 and 250, I changed all the LEN routines to IF A$ is not equal to "YES" THEN 250, or whatever. This is more apt to work on most computers, although LEN routines do work on the TRS-80 Level II. The biggest headache was with lines 1502 and 1700. I could not rearrange them to work, and David Lien (in his* A BASIC Handbook, *Compusoft Publishers) says that computed GOTOs will not work on most personal computers. I finally hit on the idea of changing the line 1700 to "IF S=1 THEN GOTO 1705", followed by lines 1702, 1703, and 1704 similarly, for S=2, 3, and 4. For V=2 thru 14, I first changed line 1500 to 1400 and changed all of the GOTO 1500 entries accordingly. Then I started with 1402 IF V=2 THEN 1510, followed with 1403 thru 1414 for the others..."*

*I applaud Mr Smartt's tenacity and patience, and ask him to accept my humble apology....***CM**

Updating the Microcomputer Listing

Since the publication of our microcomputer survey in the first issue of onComputing (Summer 1979), there have been some changes and corrections:

- *The Exidy Sorcerer computer comes with BASIC included.*
- *The OSI Superboard II and Challenger I have a 25 by 25-character video display, not 64 by 32; the price of the 8 K byte version is $418, not $598.*

(Thanks to R M Marella of Atlantic Highlands NJ for the above information.)

What's in a Name?

*In our Microcomputer Manufacturers listing (see onComputing, Summer 1979, page 102) we inadvertently referred to the MicroDaSys company as Micro Data, their former name. Our apologies....***CM**

TRS-80 Review Bug

Gentlemen:

I would like to inform you of an error in the Summer 1979 issue of onComputing. The error is on page 44 of the article "A User Reviews the Radio Shack TRS-80." Mr Shuford stated that the CLOAD? command would not work with tape number 2. I believe that he tried it with the format of CLOAD?#-2, "file-name". The correct format is CLOAD#-2,? "file-name".

James L Liles
Candor NC

Microproducts Bug

Dear Editor:

I enjoyed your first issue, (Summer 1979 onComputing) which you filled with an incredible amount of useful and interesting information. Keep up the good work.

Unfortunately, our company was overlooked in one article, "A User Reviews the Apple II System," and misrepresented in another, "Getting Started."

In the article "A User Reviews the Apple II System," Mr Cote laments that he would like to see a good assembler for the Apple II. Microproducts has had a very good assembler available for the Apple II (mentioned in an early issue of Apple's newsletter) for well over a year, and has just introduced a new and more power full 6-Character Label Editor/Assembler and also a companion 6-Character Label Disassembler/Text-File Manager. These two items are a must for any serious assembly-language programmer.

Also, Ms Hughes' article showed our Superkim singleboard, 6502-based industrial control computer with the wrong caption, "SWTP 6800, a mother board computer," on page 80.

David G Smith Jr
President
Microproducts
Redondo Beach CA

Glossary

The purpose of this glossary, a regular onComputing feature, is to define personal computing terms that might be unfamiliar to readers.

address: An identifying number (often hexadecimal or decimal) used to describe a location in computer memory.

algorithm: An orderly procedure for obtaining a particular result or solving a problem. Algorithms are often expressed in mathematical terms.

application: The use of a computer system to accomplish a specific goal. A general-purpose personal computer can be used for any one of a number of applications depending upon what programs are chosen or written by its owner.

array: In computer usage, the setting aside of a section of memory to hold a group of related data values. More specifically, arrays are indicated in the BASIC language by subscriped variables, such as A(X), where X is the subscript.

ASCII character set: The most common character-coding convention in personal computers. The name is the acronym for American Standard Code for Information Interchange. In its fullest form, it is the definition of a set of 256 different standard meanings for the 8-bit codes which can be generated by a typical computer or computer terminal. Most small computers and terminal products support only a subset of this full ASCII character-set definition, which typically includes upper- and lower-case alphabetic characters, numbers, and a set of special symbols.

assembler language: Synonym for assembly language.

assembly language: A form of computer language that uses mnemonic names to stand for one or more machine language instructions. The latter are the most basic instructions in the computer, and assembly language is a "shorthand" method for avoiding the tedious use of the long strings of ones and zeros found in machine language. The advantage of assembly language over high-level languages such as BASIC is its speed, although high-level languages are generally easier to use than assembly language.

base: The number on which a given numbering system is built. For example, the decimal number system is a 'base 10" system (ie: it has ten unique symbols or digits, and each place in a number represents a power of the base number, 10 in this case). The binary number system is base 2.

BASIC: A type of high-level language popular among personal computer users.

binary: A type of number system based on the number 2, as compared with the decimal system, which is based on the number 10. The binary system uses only the digits 0 and 1, and each place in a number represents a power of 2. For example, 101 in binary is the same as 5 in decimal.

break: A key which is used to interrupt a computation and return the computer to a user-input mode.

byte: The basic unit of information in the computer. A byte consists of binary bits, usually eight in microcomputer terms.

character: The name given to a byte of information in the personal computer when that byte is used to store a code number corresponding to one of the standard characters of, for example, the ASCII character set. A character in memory corresponds to the depression of one key on the typewriter-style keyboard of the typical computer or computer terminal.

character string: A sequence of characters (letters, numbers, and symbols) used in high-level languages such as BASIC.

COBOL: One of the standard sets of languages most often implemented on large computer systems. It is oriented toward business applications, and is beginning to make an appearance on personal computers which have a business orientation. Thus, COBOL is an acronym for common business-oriented language.

code: Synonym for a computer program, ie: a programmer generates *code.*

comment: A note incorporated in the source text of a program which is intended to make later understanding of the program easier. With languages such as assembly language, FORTRAN, or BASIC, comments are essential as a part of good programming practice. With better conceived languages like Pascal, PL/I, or COBOL, comments are less essential, but recommended whenever nonstandard and possibly obscure practices are employed. Every programming language has some form of comment statement.

computer language: An artificial language concocted for purposes of communication between human beings and computer systems. These languages range from the primitive symbolic assemblers available for most computers, through high-level, application-oriented languages including the mathematical language APL and the artificial intelligence (ie: robotics) language LISP. Most inexpensive personal computers use a relatively primitive computer language called BASIC. More expensive personal computers are delivered with software for a more powerful and simpler-to-use language called Pascal.

conditional jump: The main way of affecting the course of execution of instructions in an assembly language for some computers. It will cause execution of the program to jump to an alternate instruction location in the program, depending upon the state of one or more *flag* bits in the *condition code register* of the computer's processor. A typical example might be "jump if zero," which would cause execution to go to the specified location if the *zero* flag of the processor was *true* after the last previous operation.

cross-reference listing: Gives information about where a symbol (ie: variable, procedure, or other item) was first defined, and where it is referenced within a program. This information can often be used to great advantage when writing and debugging your own programs. When comparing two different computer systems for which you intend to write your own programs, give extra weight to the one which has cross-reference listings available for its various languages.

debug: To search for and correct mistakes in a computer program. The term is also used in reference to fixing electronic circuitry.

decimal: A "base 10" number system, the most common form of notation in Western civilization. However, since computers can calculate inherently well in a binary system, other systems are often employed which use a power of 2 as the base: binary uses 2, octal uses 2 raised to the third power, and hexadecimal uses 2 raised to the fourth power. When an external decimal representation of a number is produced from a computer program, it thus must reflect an automated binary-to-decimal conversion procedure.

diagnostic: A message sent to the user of a program or system of programs which complains about something extraordinary. In a high-level language system, a diagnostic might be the message explaining that an improper form of expression was used. In an application of a computer, a diagnostic message might be used to inform the computer user that the data just entered was inconsistent with the expected form or purposes of entering that data. The response to a diagnostic on the part of this user is to try again, using the diagnostic message as feedback to change the attempted input.

dialects: Variations on the basic theme of the computer language. Thus, BASIC is by no means the same for every system but is largely similar. Pascal, FORTRAN, and PL/I also occur in different variations on different computers.

direct memory access: A fast and convenient method of data transfer, enabling a peripheral device to transfer data directly from the memory circuits, without requiring action from the main processor (except to start the transfer, if needed). It is frequently used in video display systems and disk systems, and is often abbreviated as DMA. (See also programmed data transfer.)

disassembler: A program which takes machine language code and generates the assembler language code from which the machine language was produced. (See also assembly language.)

dot-matrix printer: Printer which employs a small array of dots to represent a course image of the characters printed. Most dot-matrix printers which print uppercase characters only use a 5 by 7 matrix of dots to represent each character of the alphabet. Printers capable of uppercase and lowercase printing typically use a 7 by 9 matrix of dots to represent a full set of alphabetic characters. The ultimate in dot-matrix technology is found in various high-resolution dot-matrix devices such as inkjet or precision impact printers which can assemble characters from matrices of 30 by 50 dots that may overlap.

dynamic memory: A type of programmable memory. Data is held in the form of electrical charges on tiny capacitors inside integrated circuits. Dynamic memory requires "refresh" operations to prevent the data from being lost. It is usually cheaper and often faster than static memory.

EBCDIC character set: The character-coding convention used in computers manufactured by the IBM corporation. The name is the acronym for Extended Binary Coded Decimal Interchange Code. Like the ASCII character set used in most small computers, EBCDIC assigns a unique interpretation to the 256 possible codes which may be stored in an 8-bit byte of information.

electrographic printing: Process in which characters are formed on the resulting printout in two stages: first the image of the printed output is transferred onto a printing surface. Then the surface is toned by the addition of opaque particles which are bonded to the surface. This technique is based on the principles of operation of the Xerox machine, and is not typically used in inexpensive printers for personal computers.

electrostatic printing: Process by which an image is made on a suitable, special-purpose conductive paper by discharging a spark between the printhead electrode and the paper. As a result of the spark, the surface layers of the paper are indelibly marked, changing the appearance from a reflective silvery color to the dark color of the underlying layers of the paper.

floating point BASIC: A type of BASIC language which allows the use of decimal numbers. The name comes from the fact that the decimal point "floats" to a new position in a number, as required, following a calculation.

font: The set of images associated with a given character set such as ASCII, EBCDIC, or the special-purpose sets used in computerized typesetting machines such as those used for this magazine. A typical font for computer output from an impact printer might be one which duplicates the font of a standard typewriter. For a low-resolution dot-matrix printer, the font might be a program in the printer's read-only memory which translates each ASCII code into a visual representation as a matrix of dots.

graphics: The techniques of creating visual images using a computer. With personal computers, graphics features use some form of black and white or color television display. Graphic displays are used to display the normal letters, numbers, and special symbols of a character set in all modern personal computers. Some personal computers have more elaborate and flexible interfaces which give the ability to draw pictures instead of using words for interactions.

handshaking: The method by which two different computer systems (or a computer and a peripheral device) coordinate communications through some form of interconnection. A key part of this process is the ability to send messages about the status of the communications link, as well as messages which are part of the intended transfer of data.

hard copy: Graphic images which get recorded on paper in a humanly readable form. Hard-copy facilities are often missing on the least expensive computers. To be most convenient, present-day computers require hard copy.

hexadecimal: A number system which uses the base 16 for its representation of integers. In computers which have fundamental units of information in memory which are byte-sized (8 bits), this provides a more convenient, external, humanly-readable representation of internal data than use of binary digits. It takes two digits selected from the numeric character set 0,1,2,3,4,5,6,7,8,9, and the six letters A,B,C,D,E,F to represent a byte of information, as opposed to eight digits to represent the same number in an external binary form. Each hexadecimal digit stands for four bits, so hexadecimal is a very natural representation to use with byte-oriented computer systems.

impact printing: The methods of making a printed image, where the paper is struck in some way, usually involving a form of ribbon as in a standard typewriter. These techniques of impact printing sometimes use dot-matrix character formation, and sometimes use predefined fonts as in the typewriter or on bands or chains of characters contained in some high-speed printers. Impact printing methods are capable of producing multiple copies at the same time, using some form of carbon paper or its equivalent.

instruction set: All of the possible op codes a microcomputer can execute. (See op code.)

integer BASIC: A type of BASIC which can process whole numbers only; no decimal numbers are allowed. (See BASIC and floating point BASIC.)

interpreter: A program which usually comes with a small computer, used to implement a high-level language such as BASIC. In an interpreted implementation of a computer language, every time a statement is executed, it must first be translated from a humanly-readable form into a form which the machine can execute. In a compiled form of the same language, this translation is done once before the program is executed. Thus, in the common interpretive forms available, the same program will run much more slowly than if the program were compiled.

interrupt: The stopping of a process so that it can be resumed at a later time.

label: A name comprised of letters, numbers, or symbols used to identify a statement or instruction in a program.

loading: The act of transferring the contents of one or more bytes of data from a memory bank into the local memory of a computer. Loading can refer to the loading of a program from a disk or tape copy into the main memory of the computer prior to execution of that program. It can also refer to the loading of a register when a few bytes are transferred from the main memory into the registers of the central processor in an assembly-language program.

location: The unique address in a particular address space of a byte of data, a record of information, or a program. For bytes of data in main memory, locations are the addresses in that main memory. For records stored on a random-access disk file, a location is the starting address of the first byte of the record in the address space of all possible bytes on the disk. For a program stored in the file structure of a disk system, its location is the choice of its name, which is looked up in a directory and automatically translated into the appropriate disk addresses.

main memory: The primary resource for storage of data and programs in a computer. It is a random-access form of memory which can be altered and changed under the control of programs. As implemented in contemporary personal computers, main memory is a transient storage area, for when power is turned off, its information content evaporates. Thus, some form of mass storage using a different technology is required if permanent records of programs and data are kept.

mass storage: A technique for keeping track of large amounts of permanently available data in a machine-readable form. Mass storage (also called bulk storage) is invariably slower in access than main memory, but yields larger potential amounts of data and permanent qualities. In small personal computers, mass storage is provided by cassette tapes or the more convenient floppy disks.

memory: A place in a computer where data and programs can be stored.

mnemonic: The name for a machine-language operation code of a computer, chosen so that the assembly-language programmer can easily remember its function based on its abbreviation. Thus, in some computers the operation "push registers onto user stack" might have the mnemonic form PSHU as a short, easily-coded abbreviation for a long, verbal description.

modem: Abbreviated form of modulator/-demodulator. A device used to send data over communication lines. The modulator section encodes the information for transmission to another modem. Incoming information is decoded by the demodulator section.

modulator: An electronic black box used to translate the television video output signals of the computer into a standard radio-frequency television signal which can then be fed into the antenna terminals of a television tuned to an appropriate channel.

object code: The machine-language form of a program, resulting from the output of the automatic translation process when using assembly language, or a high-level language compiler (as opposed to an interpreter). An object code program can be directly loaded into memory and executed, since it has already been translated from its humanly-

readable, equivalent form into the internal, executable form.

octal: A system of number representation in which the base 8 is chosen and the digits 0,1,2,3,4,5,6, and 7 are used. Many programmers prefer octal to hexadecimal notation, although octal is a natural notation of numbers only on machines whose "word size" is a multiple of three bits. On the typical, byte-oriented personal computer, use of octal for machine language is awkward because the eight bits of a byte become three octal digits grouped as 2,3, and 3 bits, versus two hexadecimal digits of a uniform size, 4 bits.

op code: Abbreviated form of operation code. A binary code used to identify a particular function which can be carried out by the computer. (Also called machine code.)

operand: A number which is operated on in a computer program.

operating system: A set of programs usually created by the manufacturer of a large computer, which allow it to be conveniently used by many different programs. In the world of small personal computers, the operating system gives the computer its "personality" as seen by the interactive user. In addition to the many manufacturer-supplied operating systems of personal computers, several machine-independent operating systems exist which can be run on many different computers. These include the Mirosoft forms of BASIC, a very traditional, large, computer-like operating system called CP/M, and the interactive, Pascal language operating system called UCSD Pascal.

Pascal: A compiled computer language that is personal computing's answer to the elaborate, conventional languages COBOL, Algol and PL/I that are often found on large systems. Pascal is an invention (circa 1970) of computer scientist Niklaus Wirth. It was initially intended as an aid to teaching computer languages. Its success as a general-purpose programming language for computer systems and applications programming has led to its widespread use in computers of every size, from the Apple II computers available in every computer store, to the world's largest and fastest supercomputer, the Cray-1.

parallel interface: A method of plugging a peripheral device into a computer such that a whole byte (or group of bytes) of data is transferred at one time. Thus, in a parallel interface, multiple wires are typically found. For a printer, a parallel interface might include seven or eight data wires and from three to five control wires. At the price of a more expensive connector, a much higher data rate results from parallel techniques compared to the alternative serial techniques.

PEEK: A BASIC instruction that allows the programmer to look at (peek at) any location in programmable memory. This instruction is often used to scan the memory locations which hold the information displayed on the video monitor in order to determine what is being displayed.

peripheral: An external device such as a printer, floppy disk unit, cassette recorder, etc, which interacts with the computer.

pin feed: A standard feature of many computer printers, to solve the problems of alignment which occur when continuous computer forms are put through a typewriter-style friction-feed platen of a printer. Problems with alignment accumulate as the paper is printed, unless a pin-feed arrangement is used with holes along both edges of the paper.

pixel: The smallest available unit of output which can be controlled by the computer in a graphics display device. In a dot-matrix printer, the pixel is one dot within the matrix; on a television display device, the pixel is one dot on the screen of the television. In ordinary printing of noncolor displays, pixels are either black or white; in color displays, each pixel can typically have one of several different colors.

POKE: A BASIC instruction used to place a value (poke) into any location in programmable memory. This instruction is often used in conjunction with the PEEK instruction to display graphics on a video display.

process control: The use of a computer (or other electrical or mechanical devices) to manipulate the parameters of a process. An example of process control is the use of a microcomputer to control home heating and air conditioning.

program: A set of instructions for a computer to execute, oftentimes the implementation of an algorithm. Programs are written in programming languages such as BASIC, FORTRAN, Pascal, COBOL, PL/I, or even a symbolic assembly language.

programmable memory: Content changeable memory, as opposed to read-only memory (the contents of which are fixed during manufacture). Programmable memory is where most programs and data are stored. It is sometimes called random access memory, or RAM, but this is a slight misnomer.

programmed data transfer: The most common method of data transfer in microcomputers. Each byte of data passes from the memory circuits, through the main processor, and out to the peripheral device in the case of output; and in the opposite direction in the case of input. Programmed data transfer requires the constant attention of the computer program for the transfer to continue, hence its name. (See also direct memory access.)

protocol: The definition of a computer communications procedure, including the hardware of the interface and the software conventions of what standard patterns will be imposed on the data being communicated.

New Products

Conducted by Charles Freiberg

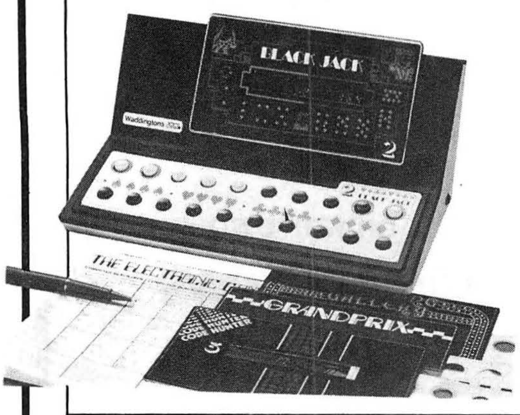
These pages are designed to keep our readers in touch with the marketplace. The material which appears here is obtained from manufacturers and is not to be taken as an endorsement by onComputing. We invite manufacturers to submit material and we publish the information we feel will be of interest to our readers.

Electronic MasterMind—A Game of Logic and Skill

An electronic version of MasterMind has been introduced by Invicta Plastics, Suite 940, 200 5th Av, NYC NY 10010. Electronic MasterMind can be played solitaire style, with two players, or with a group. Players can choose from MasterMind games in a 3, 4, or 5 digit code that are rated in degree as easy, moderate, and difficult. A player, or players, can have either a hidden random number code set by the microcomputer to solve or they may program in their own code for others to break. Clues are supplied on each try. The game also keeps track of the number of attempts made. The price of Electronic MasterMind is $30.

Circle 200 on inquiry card.

Four Computer Games Plus a Calculator



The Game Machine is an electronic game and calculator that uses a microprocessor. This desk-top computer game features four different games, including Shooting Gallery, Black Jack, Code Hunter, and Grand Prix. Additionally, it has a 4-function, 8-digit calculator. The approximate price of the Game Machine is $70. Contact Waddington's House of Games Inc, 2633 Greenleaf, Elk Grove Village IL 60007.

Circle 201 on inquiry card.

Index of Business Applications Programs

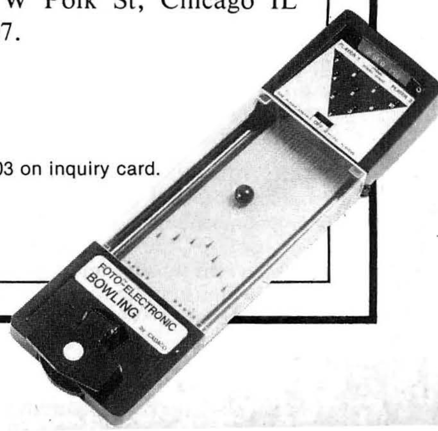
Business applications programs are indexed by application in the *Microcomputer Business Programs Index*. All listed programs are floppy disk based and each listing includes language requirements, hardware requirements, price, description, and company or author name. A list of companies addresses is provided. Currently with over seventy listings, the index is revised quarterly so that each copy delivered is up to date. The index is $8 from The Computer Store, 6526 Washington St, Yountville CA 94599.

Circle 202 on inquiry card.

Enjoy the Fun of Real Bowling

For one or two players, Foto Electronic Bowling performs all the functions of real bowling. Frame number, strike and spare, and score are shown on the light-emitting diode display. Additionally, the speed and direction of the miniature bowling ball can be controlled. The price of the Foto Electronic Bowling game is $43.90. Contact Cadaco Inc, 310 W Polk St, Chicago IL 60607.

Circle 203 on inquiry card.

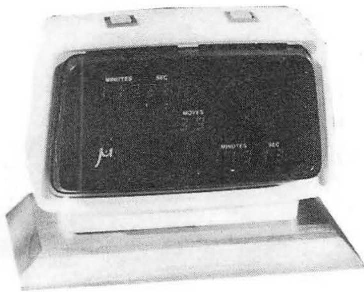


New Products

New Electronic Word Game from Texas Instruments

Designed for children eight years to adult, Mr. Challenger is a letter and word game in which players compete against the machine or against another player. The four games are: mystery word, word challenge, letter guesser and crazy letters. There are 500 words in the machine's memory, and users can enter their own words. Scoring of games is based on time, level of difficulty, and use of a "clue" key. Features include sound effects and scorekeeping. The suggested retail price for this hand-held game is $35. Contact Texas Instruments Inc, POB 53, Lubbock TX 79408.

Circle 204 on inquiry card.



Chess and Game Clock

The microprocessor controlled Micromate-180 has been designed to enhance the significance of the dimensions of time in many popular games. Although the Micromate-180 was conceived primarily for chess, it is also the perfect instrument for timing backgammon, scrabble, bridge, and basketball. The battery powered clock features a calculator-like keypad for easy programming. The Micromate-180 is priced at $199.95 delivered. Contact Micro General Corp, POB 17746, Irvine CA 92714.

Circle 206 on inquiry card.

Pascal Business Accounting Packages

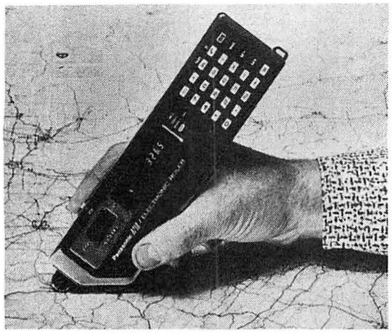
A fully integrated system of Pascal business accounting packages has been announced by P.S. Inc, 619 NP Av, Fargo ND 58102. Standardized on the UCSD implementation of the language, the software includes a general ledger package that permits a company to name and number over 1000 of its own accounts and to generate financial reports. Accounts payable, with aging and cash requirements reporting; accounts receivable with aging and sales analysis, order entry, and inventory control are all tied into the general ledger.

Circle 205 on inquiry card.

Portable Electronic Chess Computer

The Boris Diplomat is a compact, portable, battery-operated electronic chess computer. The Diplomat is capable of playing at a variety of levels of skill. The position programmer allows more advanced players to set up special board positions to practice specific strategies. Beginners use the position programmer to remove pieces for handicapping or for practice with specific positions. The Diplomat has a built-in chessboard with pieces, measures 8 by 7 by 1½ inches (20.32 by 17.78 by 3.81 cm) and operates several hours on six AA batteries or on the A/C adaptor included. The approximate retail price of the Boris Diplomat is $119.95. Contact Chafitz Inc, 1055 First St, Rockville MD 20850.

Circle 207 on inquiry card.



The Ruler That Thinks

The Panasonic Electronic Ruler/Computer uses a small displacement measuring wheel to directly measure lengths, distances, areas, and volumes in linear, square, or cubic units, in any scale, from any document. A multifunction calculator is integrated in the ruler permitting measured data to be used automatically in computations. Intermediate measurements can be stored in the calculator's memory to yield a total quantity. The computer displays values directly in millimeters, centimeters, or meters and converts to either inches or feet simply by pressing a function key.

The Electronic Ruler/Computer is priced at $99.95. Contact Chafitz Inc, 1055 First St, Rockville MD 20850.

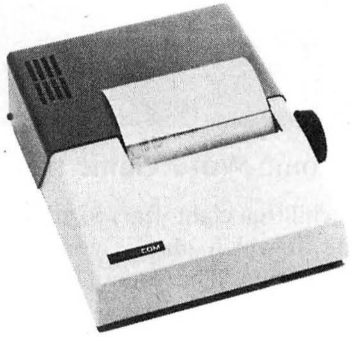
Circle 208 on inquiry card.

New Catalog Features Apple II Software

Charles Mann & Associates has announced the availability of their new catalog of software products for the Apple II computer. The catalog features full capabilities descriptions of fifty programs. The product line includes business accounting, accounts receivable, inventory, BASIC teaching, and other special purpose business applications. The applications generally use disk or tape systems and many programs are available in both of the Apple II's read-only memory languages. Contact Charles Mann & Associates, Micro Software Division, 1926 S Veteran Av, Los Angeles CA 90025.

Circle 209 on inquiry card.

Intelligent Printer for Under $400

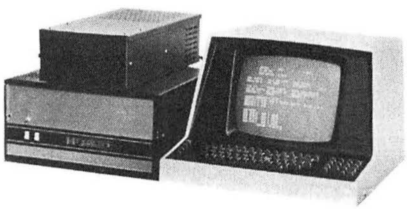


The Trendcom 100 Intelligent Printer provides the microcomputer user with 40-column hard copy on 4½ inch wide paper. Interfaces are available for the TRS-80, Apple II, PET, and Exidy Sorcerer computers. The Trendcom 100 features bidirectional 40 character per second printing with a full 96 character ASCII set, including upper and lowercase letters, numerals, and punctuation marks. The 5 by 7 dot-matrix characters are printed with either black or blue images, depending upon the paper used. This new printer uses a thick film thermal print head to eliminate wear and reliability problems.

The Trendcom 100 is fully enclosed in a metal and high impact plastic case and is available in both 115 VAC and 230 VAC versions. The price is $375. Contact Trendcom, 484 Oakmead Pky, Sunnyvale CA 94086.

Circle 210 on inquiry card.

New Microcomputer System Features Z80 Processor



Informer 3's hardware consists of a Z80 microprocessor, 48 K bytes of programmable memory, two RS-232 serial interface ports, one parallel interface port, a 2 K byte programmable read-only memory monitor, 8-inch floppy disk, and a 24 by 80 character video terminal. The software package includes Floppy BASIC (an extended disk BASIC,) diagnostics, and basic utilities, which include file copy and disk copy for either single or multiple drive systems. Present business software includes Inventory Management, Payroll, Accounts Payable and Receivable, Word Processing, Customer Mailing List, General Ledger, Program Development, and others. The Informer 3 system sells for less than $4000. Contact Digital Sport Systems, 7th and Elm Streets, W Liberty IA 52776.

Circle 211 on inquiry card.

New Products

Inventory Control and Cash Register for the Apple II

Point-of-Sale software provides business people with a convenient means of handling inventory, invoicing, back-orders, sales, and cash. While operating much like a cash register, the system maintains data on inventory level, reorder points, inventory cost, and wholesale and retail value. Sales information may be provided in the form of daily totals by month, and year to date. The balance of cash on hand is instantly available. Sales tickets, invoices, and purchase orders are automatically generated and vendor and customer files provide for easy reordering and promotional contact.

Point-of-Sale is designed for use with the Apple II with 48 K bytes of programmable memory, dual disk II drives, Applesoft II firmware card, and parallel printer. The price, including complete documentation, is $500. Contact High Technology Inc, 1611 Northwest 23rd St, Oklahoma City OK 73106.

Circle 212 on inquiry card.

A TRS-80 Disk Forecasting Package

The Business Planning Package for the TRS-80 is a floppy disk package containing a set of forecasting programs that will allow the small business user to solve a variety of business forecasting needs. The data preparation program allows the creation, modification, and deletion of disk based data sets. The data sets are accessible by all programs. Contact Applied Economic Analysis, 4005 Locust Av, Long Beach CA 90807.

Circle 213 on inquiry card.

TRS-80 Income Tax Programs

If you want to prepare your income tax on your personal computer, a full line of software for the TRS-80 32 K byte disk and line printer system is available from Contract Services Associates, 1846 W Broadway, Anaheim CA 92804. Form 1040 and related schedules are displayed on the video monitor while figures and other data are entered. Computations are performed automatically, and changes or additions can be made before printing the form.

Circle 214 on inquiry card.

Complete Business System

The SDS-100 computer is a small desk-top size system from SD Systems, POB 28810, Dallas TX 75228. It features over 1 M bytes of on-line disk storage; 64 K bytes of programmable memory; a 12 inch video screen; Z80 based processor; full keyboard and cursor control; and video field enhancements. The computer is compatible with C-BASIC, COBOL, Disk FORTRAN, Microsoft Disk BASIC, CP/M, and hundreds of compatible software programs including the SD business software family. The price of the SDS-100 computer is $7995 and it is available with 110 VAC or 220 VAC.

Circle 215 on inquiry card.

Apple Introduces a System for Smaller Businesses

This low-cost system includes the Apple II Plus computer with 48 K bytes of programmable memory, two disk drives, a video monitor, a printer, and a software package.

This system has an expanded version of BASIC built in, an automatic disk loading feature, reset protection, easy screen editing, and an audible warning that signals typing errors.

Included with this system is The Controller Business Software Package. This program is intended to meet the needs of smaller businesses because it is self-prompting and requires no programming knowledge. It consists of three program modules. The General Ledger keeps a file of up to 250 types of accounts with up to $99 milion in any one account and allows up to 1000 journal transactions each month. The Accounts Receivable maintains a file of up to 250 customer accounts and can process 1000 invoice statements per month. The system produces mailing labels, customer lists, and prints monthly account statements. The Accounts Payable maintains a file of one hundred vendors and allows 300 monthly invoices for up to $99,000 each. Checks and summaries of cash requirements by due date and vendor are printed automatically.

The system lists for $4995 and is available from Apple Computer, Inc 10260 Bandley Dr, Cupertino CA 95051.

Circle 216 on inquiry card.

Publication Lists 32 BASIC Programs for the PET

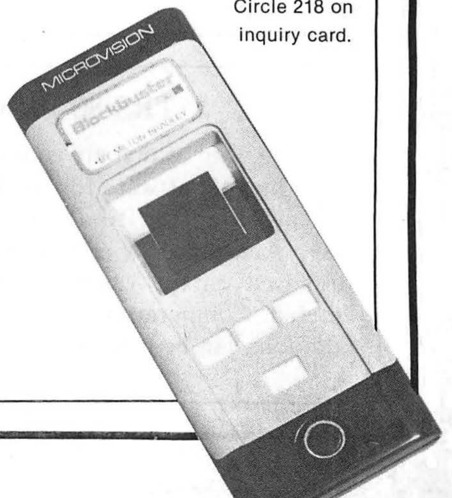
32 BASIC Programs for the PET Computer by Tom Rugg and Phil Feldman is precisely that: 32 fully documented programs ready to run on an 8 K byte Commodore PET 2001 computer. The reader has the option of making changes in these programs. This 267 page book, covering application, educational, game, graphic display, mathematical and miscellaneous programs, is priced at $15.95. Contact Dilithium Press, POB 92, Forest Grove OR 97116.

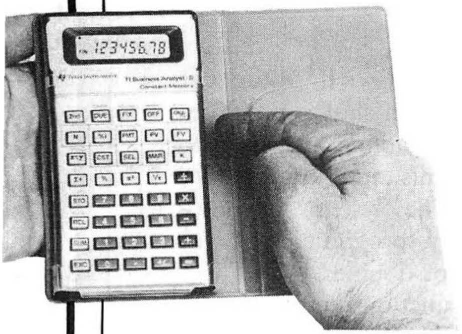
Circle 217 on inquiry card.

Microvision Features Seven Different Game Cartridges

Milton Bradley's Microvision is a hand-held mini video game with its own screen. Electronically operated, Microvision comes equipped with the game Blockbuster and six additional game cartridges may be purchased, including Bowling, Pinball, Connect 4, Star Trek Phaser Strike, Vegas Slots, and Mindbuster. Microvision is priced at $51.25 and the game cartridges range in price from $16.50 to $18. Contact Milton Bradley Co, Springfield MA 01101.

Circle 218 on inquiry card.





Financial Calculator from Texas Instruments

The Slimline Business Analyst II, a multimode financial calculator, is available from Texas Instruments Inc, POB 5012, Dallas TX 75222. It features separate operating modes for financial, statistical and profit margin computations, and also includes a constant key and nonvolatile memory that retains data whether the unit is on or off.

Special keys activate formulas for time-money and mortgage loan functions, such as compound interest, annuity payments, mortgage loans, investment yields, amortization schedules, accumulated interest, remaining balance and principal-interest split. In addition, profit margin calculations are solved with the unit's cost, sell and margin keys when two of the three values are known.

Four built-in statistical functions are provided to perform the most common statistical calculations encountered in business situations—data entry, mean, standard deviations and variance. Preprogrammed linear regression routines handle trend line analysis problems and linear projections for forecasting sales, earnings and other data. The BA-II is priced at $45.

Circle 219 on inquiry card.

Electronic Voice Response Synthesizer for TRS-80

This electronic voice response synthesizer is designed specifically as an optional attachment to Radio Shack's TRS-80 computer. Through the peripheral voice module, which comes in the form of a plug-in box, TRS-80 users will be able to program their computers to talk back to them. The Votrax synthesizer converts the output of the TRS-80 computer into electronically synthesized speech with an unlimited vocabulary and will interface with the computer without any modifications.

Included with the purchase of the TRS-80 voice response unit is a Blackjack program that makes card shuffling sounds, asks for your bets, verbally prompts you on your turn and makes humorous comments about your bets, winnings and losings. Other programs capable of being produced by home users of the Votrax voice module are available.

For further information, contact Votrax, Division of Federal Screw Works, 500 Stephenson Hwy, Troy MI 48084.

Circle 220 on inquiry card.

Play Tennis Against a Computer

Dedicated tennis players can now practice their technique against a computer-controlled mechanical opponent. This device, known as the System, can be electronically programmed for any sequence of 1 to 99 shots to different court locations, with different delays. Each shot is programmable, enabling the user to play tennis, not just practice shots. The System can automatically program itself for singles or doubles.

Six-function remote control gives fingertip control from anywhere on the court. The telescoping head fires serves and smashes from an overhead height of eight feet, and groundstrokes from four feet. This is all done using an 80 switch pressure sensitive touch panel and an advanced microprocessor with a built-in tape recorder. Contact United States Machine Works Inc, 21 Williams Pl, Lansdale PA 19446. Circle 222 on inquiry card.

Computer Weaving

The Video Loom program uses an Apple II to simulate handloom weaving, with as many as seventy harnesses on a color video display. The computer asks the weaver for information about how the loom is to be set up and the weaver responds by typing answers on an ordinary typewriter keyboard. Once the loom is warped and the weaving sequence is quickly established, the computer weaves a full color print of the design. The weaver can make changes in the pattern before photographing or storing the pattern on a floppy disk.

This program is available on a 5 inch floppy disk for use on any Apple II with at least 32 K bytes of memory. $49.95 from Systems for the Arts, 1510 Grant St, Berkeley CA 94703. Circle 223 on inquiry card.

Use Your Computer to Manage an Apartment Building

An apartment complex management program is available for the Radio Shack TRS-80 and Micropolis Mod II systems. The package includes extensive documentation and a user's guide written for non-programmers. The system includes an initialization segment and forms used to organize and set up the data base. Functions include listing transactions, posting rents, listing vacated tenants, tabulation of all transactions, vacancy listing, delinquent tenant listing, mailing labels, and more.

The price of the program is $150. Contact Honest John's Software, 8929 Cardinal Ter, Brentwood MO 64144. Circle 224 on inquiry card.

Home Study Microcomputer Courses

Integrated Computer Systems, 3304 Pico Blvd, POB 5339, Santa Monica CA 90405, has developed a series of home study courses on microcomputers. The courses include a beginner oriented, 8080A based program which covers software and hardware information. The advanced continuation course teaches microcomputer interfacing through a wide range of devices including analog to digital (A/D) and digital to analog (D/A) converters, programmable timers and interrupt circuits, opto couplers and audio cassette interfaces.

Each course includes a 700 page workbook and fully assembled and tested hardware on which to perform the exercises. The prices are $695 for the software and hardware course and $595 for the interfacing course, or $1195 for both courses purchased together. Circle 225 on inquiry card.

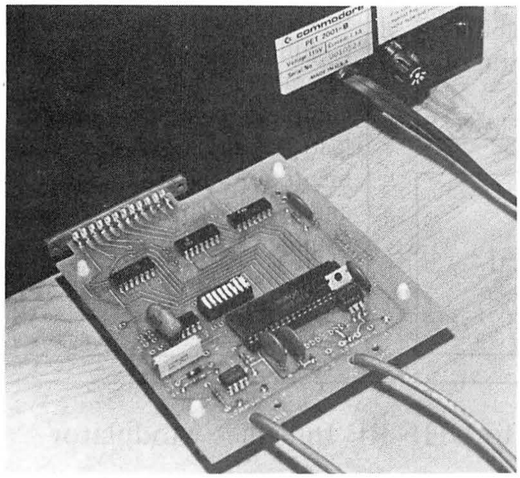
New Products

Computerized Carpooling System

CarShare is a sophisticated carpooling information system that is being utilized in Houston Texas. When a phone call is received from a CarShare candidate, data concerning their name, work address, home and work phone numbers, work days, work time, and whether the request is to share their vehicle or to ride is entered into a computer. The computer searches for a candidate living within 3 miles and working within 1.5 miles of the requester. The computer then prints a letter of reply with the information for mailing to the candidate. Calls are averaging about one hundred a day and businesses in the Houston area are planning direct video hookup with the CarShare system.

Contact Datapoint Corp, 9725 Datapoint Dr, San Antonio TX 78284.

Circle 226 on inquiry card.



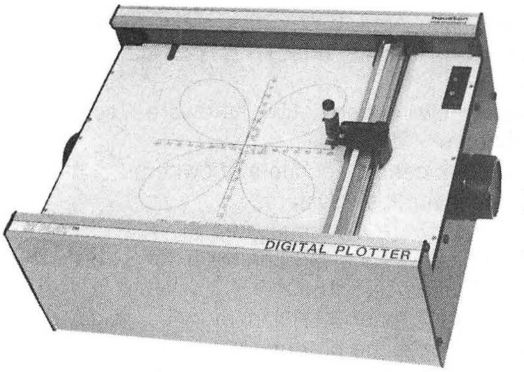
RS-232 Printer Adapter for the Commodore PET

A line of peripheral adapters for the Commodore PET is available from Connecticut microComputer, 150 Pocono Rd, Brookfield CT 06804. The Pet ADA Model 1200 drives an RS-232 printer from the PET IEEE-488 bus. The adapter allows the PET owners to obtain hard copy program listings and to type letters, manuscripts, mailing labels, tables of data, etc, using a printer with a standard RS-232 interface.

The PET ADA Model 1200 is available assembled and tested without power supply, case or RS-232 connector for $98.50, or complete for $169. Add $5 for shipping and handling. Specify data transmission rate when ordering (300 bps is supplied unless otherwise requested).

Circle 227 on inquiry card.

New Products



Digital Plotter from Houston Instrument

For the personal computer user interested in graphic arts applications, this digital plotter, called the Microplotter 2, is available from Houston Instrument, One Houston Sq, Austin TX 78753. The resolution of the pen tip is either 0.005 inches or 0.01 inches giving a total of 1400 by 2000 picture elements or 700 by 1000 picture elements, respectively, in its 7 by 10 inch (17.8 by 25.4 cm) plotting area. Standard ink cartridges are available in four colors, allowing the user to switch cartridges to produce mixed color plots. It is also possible for the individual user to adapt a standard drafting pen to the plotter, allowing a much wider selection of colors to be used in the form of personally chosen inks. The Microplotter 2 is priced at $1085.

Circle 230 on inquiry card.

Light Pen Compatible with Most Minicomputers and Microcomputers

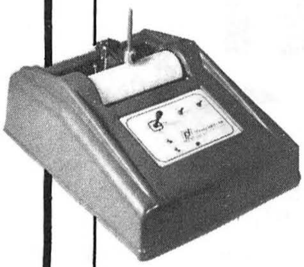
This light pen requires only three connections and is compatible with most minicomputer and microcomputer systems. The pen, called Vidiet-stik, has many uses: games, educational drills for preschoolers, "menu" selection, joystick substitution, and electronic music applications.

Vidiet-stik software is compatible with all 8080 and Z-80 computer systems and adaptable to other systems.

Each unit sells for $39.95 (plus $1.50 for postage and handling) and includes the assembled and tested pen, complete computer interface instructions, operating documentation, as well as checkout and game software listings. Contact Esmark Inc, 507½ E McKinley Hwy, Mishawaka IN 46544.

Circle 232 on inquiry card.

Interface Options for Anderson Jacobson Terminal



The Anderson Jacobson AJ841 terminal is available with an RS-232 serial interface for use in connecting the unit to a personal computer. Data can be transmitted at rates of from 110 to 1200 bits per second.

The terminal is also available with a parallel interface for connection to any computer using the S-100 bus standard. For further information contact Anderson Jacobson Inc, 521 Charcot Av, San Jose CA 95131.

Circle 231 on inquiry card.

Computer Video to UHF RF Interface Modulator

The Micro-Verter is designed to interface microcomputers to color or black and white television sets. It operates in the UHF channels above channel 14, beyond the normal range of switching harmonics, and is designed to interface directly with the Apple II as well as with most other microcomputers. It comes complete with video cable and radio frequency (RF) output coupler and requires no direct connection to antenna terminals except in special cases. The radio frequency signal is coupled directly into the UHF tuner input via a 1 cm coupler on the back of the modulator. The approximate size of the unit is 2 by 3.5 by 4.5 inches (5.5 by 8.5 by 11.5 cm) and it is priced at $35. For more information contact ATV Research, 13th and Broadway, Dakota City NE 68731.

Circle 233 on inquiry card.

New Products

A Compact Extension Terminal

The TRANSACTOR III data terminal includes a single line 32 character gas discharge display and a 53 key Teletype style keyboard. It can be directly attached to any computer with an RS-232 or 20 mA current loop interface or can be attached to a communications line through a modem.

The price is $995 from Computerwise Inc, 4006 E 137th Ter, Grandview MO 64030.

Circle 228 on inquiry card.

How to Find the Personal Computer You Want

A 24 page publication entitled *Personal Computers for the Businessman* explains what a personal computer is and how it differs from a minicomputer. It also describes a shopping strategy to follow when the decision is made to purchase a system. A major part of the report is an overview of the best known manufacturers currently in the personal computer market. The configuration of each system is given as well as the price range. There is also a list of manufacturers and suppliers in the back of the publication. The report costs $7.50. Contact Management Information Corp, 140 Barclay Ctr, Cherry Hill NJ 08034. ■

Circle 229 on inquiry card.

Coming Events

December, Datapro Seminars, Datapro Research Corp, 1805 Underwood Blvd, Delran NJ 08075 is holding various seminars covering such topics as data communication, distributed systems, data processing, word processing, minicomputers and small business computers, and more. The two-day seminars will be held in major cities around the US. For information, call toll-free: 800-257-9406.

December 3-5, COMDEX '79, MGM Grand Hotel, Las Vegas NV. This conference and exposition for third party sellers of computer systems, word processing systems, peripherals and software packages and media will focus on solutions to business problems normally encountered in structuring a successful dealership and the operational aspects of the dealership from both the supplier and customer side. Contact The Interface Group, 160 Speen St, Framingham MA 01701.

December 8-9, Data Processing for Businesspeople, Cherry Hill Inn, Cherry Hill NJ. Management Information Corporation presents this seminar to meet the needs of company management in understanding computers. The seminar includes basic concepts of data processing alternatives (service bureaus, timesharing), small business computer systems, program packages availability and selection, managing the computer system, and the future of data processing. Contact Management Information Corporation, 140 Barclay Center, Cherry Hill NJ 08034.

December 10-11, New York NY and December 13-14, Washington DC, New User Documentation Workshops. These two day workshops will focus on how to write good DP user manuals with emphasis on analysis of specific user needs, planning and outlining, effective writing, illustration and packaging of documentation. The program includes lectures on basic concepts followed by small group discussions. Contact Progressive Communications Inc, The Alamo 310, 128 S Tejon St, Colorado Springs CO 80903.

December 10-12, Project Management for Computer Systems, Chicago IL. This seminar will illustrate techniques for planning, implementing, installing, and controlling projects. Contact The University of Chicago, 1307 E 60th St, Chicago IL 60637.

January 15-18, TV-Microelectronics and Microprocessing Exhibition, National Exhibition Centre, Birmingham, England. Manufacturers and suppliers of microprocessors, electronic and microcomputer games, video display units,video cameras, projection systems, and digital consumer electronics are invited to participate. Over 9000 retailers, wholesalers, distributors and government buying authorities are expected to attend this show. For more information, contact TMAC, 680 Beach, Suite 428, San Francisco CA 94109.

January 23-26, International Microcomputers Minicomputers Microprocessors (IMMM), Harumi Exhibition Centre, Tokyo Japan. This is a show for manufacturers, commercial and financial establishments, service industries and institutions, and design engineers interested in buying computer systems, components and services. For more information, contact Industrial and Scientific Conference Management Inc, 222 West Adams St, Chicago IL 60606.

coming in onComputing

Ohio Scientific's New Color Graphics Computer: OSI breaks into the multicolor race: a look at the C4P, an updated version of the Challenger 2 4-P.

Small Business Systems: How to get started; Finding the right computer store and the best consultant. How much should you pay?

The TRS-80 in the Classroom: One forward thinking school in Oregon has introduced personal computers into their entire curriculum.

The Small Town Doctor and the Personal Computer: Poynette, Wisconsin is the unlikely location for a small revolution in computer medicine.

Plus the regular onComputing features.

interAction

Article Number		Page Number
1	Computer Games and Toys	6
2	Use Your Pocket Calculator to Save Money	12
3	Understanding Personal Computer Software	17
4	Sargon vs Microchess	26
5	The Talcott Mountain Science Center	34
6	A Voice for Bill	48
7	The Personal Check Writer	59
8	Produce evaluation: The Heathkit H8	65

Reader Service

Inquiry Number		Page Number
5	Atari	5
10	Automated Simulations	93
15	Buss	93
20	BYTE Books	49
*	Camelot Direct	19
30	Centronics	23
35	COMPRINT (Computer Printers Int'l)	56, 57
40	CompuServe (MicroNet)	15
45	Computer Nook	77
50	Dr. Daley	63
55	Disco-Tech	55
60	H & E Computronics	47
65	Hayden Book Publishing	89
70	DC Hayes	73
72	Heath Co.	C III
80	Home Computers	63
85	Micro DaSys	85
90	Micro Music	35
95	NRI Schools	33
100	Ohio Scientific Instrument	CIV
105	onComputing	16
*	onComputing Wats Line	30
110	PerCom Data	CII
115	Personal Software	1
120	RCA	91
*	St. Jude	30
125	Terrapin	77
*	Whales	25

*Please correspond directly with the company

To get further information on the products advertised in onComputing, fill out the reader service card with your name and address. Circle the appropriate numbers for the advertisers you select from the list, and drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by onComputing.

READER SERVICE

Winter 1979

For fastest service transfer mailer label from wrapper to coupon provided below. Requests cannot be honored unless zip code is given. This card valid for 90 days only.

NOTE—If label is missing or defaced fill out coupon carefully—PLEASE PRINT—this is the only way to get requested material to you.

Name ______________________________

(Title) ______________________________

(Company) ______________________________

Address ______________________________

City ______________ State __________ Zip Code __________

1	21	41	61	81	101	121	141	161	181	201	221	241	261	281
2	22	42	62	82	102	122	142	162	182	202	222	242	262	282
3	23	43	63	83	103	123	143	163	183	203	223	243	263	283
4	24	44	64	84	104	124	144	164	184	204	224	244	264	284
5	25	45	65	85	105	125	145	165	185	205	225	245	265	285
6	26	46	66	86	106	126	146	166	186	206	226	246	266	286
7	27	47	67	87	107	127	147	167	187	207	227	247	267	287
8	28	48	68	88	108	128	148	168	188	208	228	248	268	288
9	29	49	69	89	109	129	149	169	189	209	229	249	269	289
10	30	50	70	90	110	130	150	170	190	210	230	250	270	290
11	31	51	71	91	111	131	151	171	191	211	231	251	271	291
12	32	52	72	92	112	132	152	172	192	212	232	252	272	292
13	33	53	73	93	113	133	153	173	193	213	233	253	273	293
14	34	54	74	94	114	134	154	174	194	214	234	254	274	294
15	35	55	75	95	115	135	155	175	195	215	235	255	275	295
16	36	56	76	96	116	136	156	176	196	216	236	256	276	286
17	37	57	77	97	117	137	157	177	197	217	237	257	277	297
18	38	58	78	98	118	138	158	178	198	218	238	258	278	298
19	39	59	79	99	119	139	159	179	199	219	239	259	279	299
20	40	60	80	100	120	140	160	180	200	220	240	260	280	300

interAction

Winter 1979

interAction is your direct line to the editor's desk. Each month, the two top rated authors receive bonuses based on your votes. To use this card, refer to the list of authors, titles, and corresponding article numbers on the opposite page. Then rate each article on a scale from 0 to 10 below by circling the appropriate rating number to the right of each article number. Your feedback helps us produce the best possible magazine each month.

.... Rating ....

ARTICLE NUMBER		POOR		FAIR	GOOD		VERY GOOD			EXCELLENT	
1	0	1	2	3	4	5	6	7	8	9	10
2	0	1	2	3	4	5	6	7	8	9	10
3	0	1	2	3	4	5	6	7	8	9	10
4	0	1	2	3	4	5	6	7	8	9	10
5	0	1	2	3	4	5	6	7	8	9	10
6	0	1	2	3	4	5	6	7	8	9	10
7	0	1	2	3	4	5	6	7	8	9	10
8	0	1	2	3	4	5	6	7	8	9	10
9	0	1	2	3	4	5	6	7	8	9	10
10	0	1	2	3	4	5	6	7	8	9	10
11	0	1	2	3	4	5	6	7	8	9	10
12	0	1	2	3	4	5	6	7	8	9	10
13	0	1	2	3	4	5	6	7	8	9	10
14	0	1	2	3	4	5	6	7	8	9	10
15	0	1	2	3	4	5	6	7	8	9	10
16	0	1	2	3	4	5	6	7	8	9	10
17	0	1	2	3	4	5	6	7	8	9	10
18	0	1	2	3	4	5	6	7	8	9	10
19	0	1	2	3	4	5	6	7	8	9	10
20	0	1	2	3	4	5	6	7	8	9	10

Comments ______________________________

____

____

____

onComputing™

Place
Stamp
Here

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS Permit Number 7976 New York NY 10001

POSTAGE WILL BE PAID BY ADDRESSEE

onComputing
70 Main St
Peterborough NH 03458
USA

onComputing
POB 1911 GPO
New York NY 10001
USA

onComputing™

no loose ends

Computer, floppy, smart terminal, 16K RAM. $1595*

H89 All-In-One Computer

Heath takes the risk out of selecting a balanced computer system. Now, video terminal, floppy, keyboard and 8-bit computer are brought together in one self-contained, compact unit. Nothing hangs out.

Two Z80's

The personal computer has never been simpler. Or smarter. Two Z80 microprocessors mean terminal never shares power with computer, as do most desk-top units. So this terminal is capable of a multitude of high-speed functions, all controllable by keyboard *or* software.

100K bytes storage

Built-in floppy disk system gives you fast access to programs and data. Each 5¼-inch diskette has 100K bytes of storage area, enough to hold entire files. The All-In-One comes with 16K RAM, expandable to 48K.

Innovative software

The All-In-One Computer runs programs written in MICROSOFT™ BASIC and Assembly Languages. And it accepts all current software written for the popular Heathkit H8 Computer, including innovative Word Processing and Business Accounting Software.

Learn by building

What better way to learn about computers than to build one yourself? Like all Heath electronic kits, it comes to you with its own easy-to-follow assembly manual and a nationwide network of service centers to assure smooth sailing.

FREE CATALOG

For complete details on the Heathkit H89 All-In-One Computer and nearly 400 other electronic kits for your home, work or pleasure, send today for the latest Heathkit Catalog of values.

*$1195 without floppy. Mail order kit price, F.O.B. Benton Harbor, MI. Heathkit Products are also sold and serviced at Heathkit Electronic Centers in major cities throughout the U.S. See your white pages. Prices subject to change without notice.

Heathkit®

CP-176

HEATH COMPANY, DEPT. 013-590, BENTON HARBOR, MI 49022

Circle 72 on inquiry card.

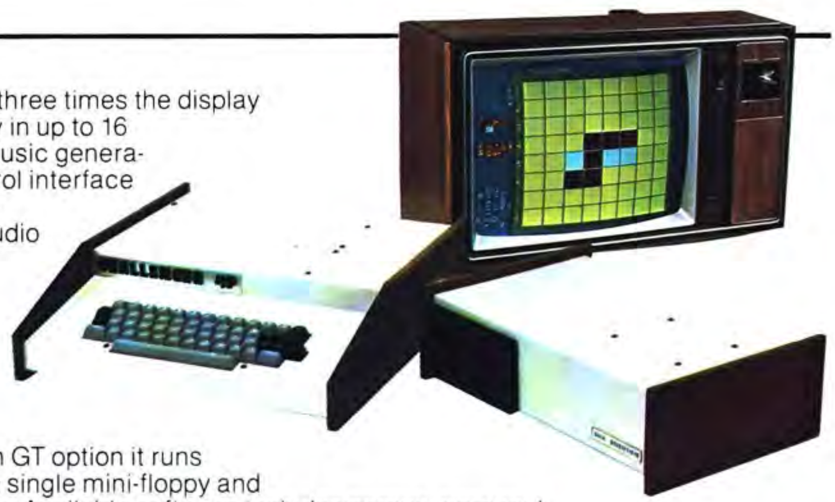
The Personal Computer Line by OHIO SCIENTIFIC

Personal Computers

C1P: $349 A dramatic breakthrough in price and performance. Features OSI's ultra-fast BASIC-in-ROM, full graphics display capability, and large library of software on cassette and disk, including entertainment programs, personal finance, small business, and home applications. It's a complete programmable computer system ready to go. Just plug-in a video monitor or TV through an RF converter, and be up and running. 15K total memory including 8K BASIC and 4K RAM—expandable to 8K.

C1P MF: $995 First floppy disk based computer for under $1000! Same great features as the C1P plus more memory and instant program and data retrieval. Can be expanded to 32K static RAM and a second mini-floppy. It also supports a printer, modem, real time clock, and AC remote interface, as well as OS-65D V3.0 development disk operating system.

Professional Portables



C4P: $698 The professional portable that has over three times the display capability of C1Ps. Features 32 x 64 character display in up to 16 *colors,* graphics, audio output, a DAC for voice and music generation, key pad and joystick interfaces, AC remote control interface and much more. Utilizes a 4-slot BUS (2 used in base machine), 8K BASIC-in-ROM, 8K of *static* RAM and audio cassette interface. Can be directly expanded to 32K static RAM and two mini-floppy disks.

C4P MF: $1695 The ultimate portable computer has all the features of the C4P plus real time clock, home security system interface, modem interface, printer interface, 16 parallel lines and an accessory BUS. The standard machine operates at twice the speed of currently available personal computers (with GT option it runs even faster!). The C4P MF starts with 24K RAM and a single mini-floppy and can be directly expanded to 48K and two mini-floppies. Available software includes games, personal, business, educational and home control applications programs as well as a real time operating system, word processor and a data base management system.

*Monitors and cassette recorders not included. Ohio Scientific offers a combination TV/Monitor (AC-3P) for $115.

Home/Small Business Systems

C8P: $895 Same great features as the C4P in a tremendously expandable "mainframe package." Features over three times the expansion capability of the C4P for advanced home and demanding business applications. Can be expanded to 48K RAM, dual 8″ floppies, hard (Winchester) disks and multiple I/O devices such as Voice I/O and a universal telephone interface.

C8P DF: From $2597 The ultimate Home/Very Small Business Computer at a personal computer price. Features 32K RAM (expandable to 48K) and dual 8″ floppy disks (stores eight times as much information as a mini-floppy). Has all personal computer capabilities including 32 x 64 display, color graphics, sound, DAC, joystick interfaces, home features including real time clock, AC remote interface, home security and fire detection interface and can be expanded to include voice I/O and a universal telephone system for answering and initiating calls! Its large memory capability and 8″ floppies allow it to run most Ohio Scientific business system software including a compete accounting system, word processor and information management system.

The C8P DF is designed to be the "Brains" of the home of the future and the small business office of the future!

OHIO SCIENTIFIC

1333 S. CHILLICOTHE RD., AURORA, OHIO 44202 (216) 562-3101

Circle 100 on inquiry card.